A GROUP THEORETIC TABU SEARCH METHODOLOGY

FOR SOLVING THE THEATER DISTRIBUTION VEHICLE

ROUTING AND SCHEDULING PROBLEM

DISSERTATION

John R. Crino, Major, USA

AFIT/DS/ENS/02-02

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

Report Documentation Page

Report Date	Report Type	Dates Covered (from... to)
18 Jun 02	Final	Apr 01 - May 02

Title and Subtitle A Group Theoretic Tabu Search Methodology for Solving the Theater Distribution Vehicle Routing and Scheduling Problem	**Contract Number**
	Grant Number
	Program Element Number
Author(s) Major John Richard Crino, USA	**Project Number**
	Task Number
	Work Unit Number
Performing Organization Name(s) and Address(es) Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/EN) 2950 P Street Bldg 640 WPAFB OH 45433-7765	**Performing Organization Report Number** AFIT/DS/ENS/02-02
Sponsoring/Monitoring Agency Name(s) and Address(es) AFOSR/NM Attn: Maj Juan Vasquez 801 N. Randolph St., Rm 732 Arlington VA 22203-1977	**Sponsor/Monitor's Acronym(s)**
	Sponsor/Monitor's Report Number(s)

Distribution/Availability Statement
Approved for public release, distribution unlimited

Supplementary Notes
The original document contains color images.

Abstract
The application of Group Theory to Tabu Search is a new and exciting field of research. This dissertation applies and extends some of Collettis (1999) seminal work in group theory and metaheuristics in order to solve the theater distribution vehicle routing and scheduling problem (TDVRSP). This research produced a robust, efficient, effective and flexible generalized theater distribution model that prescribes the routing and scheduling of multi-modal theater transportation assets to provide economically efficient time definite delivery of cargo to customers. In doing so, advances are provided in the field of group theoretic tabu search and its application to difficult combinatorial optimization problems, e.g., the multiple trip multiple services vehicle routing and scheduling problem with hubs and other defining constraints.

Subject Terms
Group Theory, Symmetric Group on n-letters, Metaheuristics, Tabu Search, Vehicle Routing Problems, Vehicle Scheduling Problems, Vehicle Routing and Scheduling Problems, Theater Distribution

Report Classification unclassified	**Classification of this page** unclassified
Classification of Abstract unclassified	**Limitation of Abstract** UU

Number of Pages
366

AFIT/DS/ENS/02-02

A GROUP THEORETIC TABU SEARCH METHODOLOGY FOR SOLVING THE THEATER DISTRIBUTION VEHICLE ROUTING AND SCHEDULING PROBLEM

DISSERTATION

Presented to the Faculty

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

in Partial Fulfillment of the Requirements for the

Degree of Doctor of Philosophy

John R. Crino, B.S., M.S.,
Major, USA

June 2002

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

AFIT/DS/ENS/02-02

A GROUP THEORETIC TABU SEARCH METHODOLOGY

FOR SOLVING THE THEATER DISTRIBUTION VEHICLE

ROUTING AND SCHEDULING PROBLEM

John R. Crino, B.S., M.S.
Major, US Army

Approved:

Date

____________________________________ ____________
James T. Moore (Chairman)

____________________________________ ____________
J. Wesley Barnes (Member)

____________________________________ ____________
COL William P. Nanry (Member)

____________________________________ ____________
Lt Col Raymond R. Hill (Member)

____________________________________ ____________
Mark E. Oxley (Member)

____________________________________ ____________
Kirk A. Mathews (Dean’s Representative)

Accepted:

______________________________ ____________
Robert A. Calico, Jr. Date
Dean, Graduate School of Engineering and Management

Acknowledgments

I would like to express my sincere appreciation to my faculty advisor, Dr. James Moore, for his guidance and support throughout the course of this research effort. I would also like to thank my committee members Dr. Barnes, COL Nanry, Dr. Oxley, and Lt Col Hill for their time and effort in guiding my research to a successful completion.

Special thanks go to Dr. Bruce Colletti, Lt Rob Harder, Capt Vic Wiley and Capt Todd Combs. Dr. Colletti spent numerous hours assisting with the group theory portions of the research. His ability to translate my "rough" concepts into algebraic elegance was invaluable. Lt Harder assisted with the operation of his tabu search architecture and made modifications to meet my requirements. Capt Wiley responsively modified his Java™ Group Classes to meet the needs of my research efforts. Capt Combs, my office mate, provided effective synergy throughout the three years with coursework, research, coding, and meetings at Prairie Trace.

Finally, considerable love and appreciation go to my wife and children, who stood by me throughout this endeavor. Their patience, understanding, and sacrifice allowed me the time and effort required to complete this arduous venture.

John R. Crino

Table of Contents

Page

Page

Page

Page

List of Figures

Figure Page

Figure Page

List of Tables

Table Page

Table Page

Glossary of Terms

AF	Air Force
AO	area of operation
APOD	aerial port of debarkation
APOE	aerial port of embarkation
ASP	ammunition supply point
ATP	ammunition transfer point
BSA	brigade support activity
C-17	strategic and theater cargo aircraft
C-130	theater cargo aircraft
CINC	commander in chief
CSA	corps support activity
DSA	division support activity
ETDD	early time definite delivery
FIFO	first-in first-out
GTTS	group theoretic tabu search
GVRP	generalized vehicle routing and scheduling problem
H	Hub, a transshipment node
ISB	intermediate staging base
LOC	lines of communication
MOG	maximum on the ground capacity (either working or parking)
MOGA	maximum on the ground capacity for aircraft
MOGG	maximum on the ground capacity for ground vehicles
MTMS	multiple trip multiple services TDVRSP instance
MTMS w/hub	multiple trip multiple services with hub TDVRSP instance
MTW	multiple no delivery time windows

POD	port of debarkation
SA	support activity
S_n	symmetric group on n-letters
SPOD	seaport of debarkation
SSA	supply support activity
TAA	tactical assembly area
TDD	time definite delivery
TDMS	theater distribution management system
TDVRSP	theater distribution vehicle routing and scheduling problem
Tier	denotes the collection of customers served by a particular depot or hub
TPFDD	time phased force deployment document
TSA	theater support activity
TTP	trailer transportation point
VRSP	vehicle routing and scheduling problem

List of Symbols and Operators

$\oplus$	fixed binary operation
π	permutation of letters
$\leq$	sub-group
$<$	proper sub-group
$< >$	represents a group via its generators.
$\subseteq$	sub-set
$\subset$	proper sub-set
*	product
C	set of customer letters
C_i	sub-set of customer letters
CClass (G, g)	conjugacy class of $g \in G$, i.e. $\{g^h : h \in G\}$.
CC*k*	a labeling system for CClass(S_n, x) $\forall x \in S_n$
Cent (H, g)	the H-elements that commute with g, i.e. $gh = hg$ or $h^{-1}gh = g^h$.
DCosets (G, J, K)	double cosets, i.e. the set $JgK = \{jgk: j \in J, k \in K\}$
$fix(g)$	the number of X-elements that g fixes, i.e. $x^g = x$.
G, H	groups G or H
$_GX$	group action of group G on set X
mov(p)	letters moved by permutation p
$N_i(p)$	the set of all non-null cyclic-positions in p of all C_i -letters.
Orbit (G, x)	G orbit of x, i.e., $x^G \equiv \{x^g: g \in G\}$.
$p \oplus move$	product of permutation p and *move*

p^{move}	permutation p conjugated by *move*
$(p \oplus move, p^{move})$	a solution created either by $p \oplus move$ or p^{move}
S_n	symmetric group on n-letters
V	set of vehicle letters
X	a non empty set of permutations
$X(k)$	the k different cyclic form structures of X
y^k	conjugation of y by k, i.e. $k^{-1} \oplus y \oplus k$
yx	same notation as $y \oplus x$

AFIT/DS/ENS/02-02

Abstract

The application of Group Theory to Tabu Search is a new and exciting field of research. This dissertation applies and extends some of Colletti's (1999) seminal work in group theory and metaheuristics in order to solve the theater distribution vehicle routing and scheduling problem (TDVRSP).

This research produced a robust, efficient, effective and flexible generalized theater distribution model that prescribes the routing and scheduling of multi-modal theater transportation assets to provide economically efficient time definite delivery of cargo to customers. In doing so, advances are provided in the field of group theoretic tabu search and its application to difficult combinatorial optimization problems, e.g., the multiple trip multiple services vehicle routing and scheduling problem with hubs and other defining constraints.

A GROUP THEORETIC TABU SEARCH METHODOLOGY

FOR SOLVING THE THEATER DISTRIBUTION VEHICLE

ROUTING AND SCHEDULING PROBLEM

I. Introduction

This dissertation is a presentation of a group theoretic tabu search approach applied to a generalized military theater distribution problem. The military theater distribution problem is described in Section 1.1. The military interest for modeling the theater distribution problem is presented in Section 1.2. The group theoretic tabu search methodology used to solve the theater distribution problem is introduced in Section 1.3 and further described in Chapter 3. Section 1.4 presents the theater distribution problem as a vehicle routing and scheduling problem.

1.1 Theater Distribution Problem Description

Military theater distribution of cargo and personnel is a complex and multifaceted operation. Numerous Department of Defense manuals present the many aspects of military theater distribution. This section is by no means a comprehensive discussion of the military theater distribution operation. Instead, an overview is presented to motivate the modeling efforts of this research.

1.1.1 Definition of Theater Distribution

Theater distribution is the flow of personnel, equipment, and materiel within theater to meet the geographic combatant commander's intent (JP 4-01.4, 2000). The "theater" is a geographical location outside the continental United States for which a commander of a unified command is assigned military responsibility (JP 1-02, 1996). The theaters of operation refer to those portions of an area of war necessary for military operations (FM 100-10-1, 1999). If there are multiple major threats in a theater, then subordinate theaters or areas of operations (AO) may be designated. For the purpose of this research, a theater is a geographic location, under the control of a joint task force or combatant commander.

A distribution or "logistics" pipeline is a channel through which the Department of Defense conducts distribution operations. The distribution pipeline is divided into two parts: strategic and theater. The strategic portion of the pipeline is the flow of logistical support from the points of origin or sources of support external to a supported theater (JP 4-01.4, 2000). The theater portion of the pipeline, which is the interest of this research, comprises all the networks within theater through which cargo and personnel flow before reaching their final destination (JP 4-01.4, 2000). This is displayed in Figure 1.1 below, where the strategic pipeline is represented by the flow of logistical support from the continental United States, Europe, and pre-positioned stocks. Cargo and personnel are delivered to the theater via airlift and sealift. The theater pipeline begins at the Aerial Ports of Debarkation (APODs) and Sea Ports of Debarkation (SPODs), where cargo and personnel are received from strategic lift assets. The cargo and personnel are staged and

then moved through a series of transshipment points (Hubs) to tactical assembly areas within the theater.

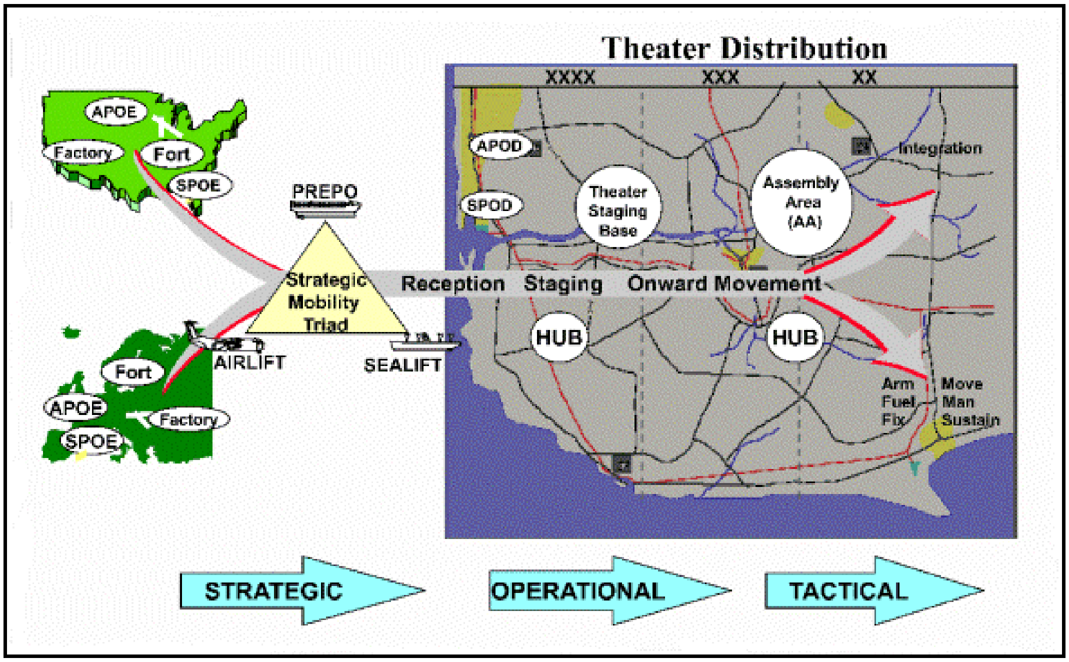

Figure 1.1 Distribution Pipeline (FM 100-10-1)

The responsible agent for theater distribution is the supported geographical combatant commander. The primary units responsible for the execution of the theater distribution mission are the combatant commander's logistics staff, the Service component logistics staff, and the Service component operational units, which are linked to operate and perform the day-to-day distribution functions (JP 4-01.4, 2000).

The physical network of the theater distribution pipeline consists of fixed structures and established facilities available to support distribution operations. These include airfields, seaports, roads, railroads, inland waterways (IWW), pipelines, terminals, road and railroad bridge/tunnels, and buildings. Within a theater, these are all sensitive to available host nation infrastructure (FM 100-10-1, 1999).

There are two primary phases of theater distribution: force projection and force sustainment. Force projection is the initial deployment of forces into theater. This process usually is dictated by the Time Phased Force Deployment Document (TPFDD), and includes all actions required within the distribution system to assemble deploying elements such as personnel, equipment, unit cargo, and materiel stocks into an operationally capable force (FM 100-10-1, 1999). As the theater matures, the distribution pipeline efforts shift to force sustainment. Sustainment is the supply of logistics to sustain the operational force in theater.

1.1.2 Theater Distribution Components

The theater distribution system is a complex of facilities, installations, methods, and procedures designed to receive, store, maintain, distribute, and control the flow of military materiel between the point of receipt into the military system and the point of issue to using activities and units (JP 1-02, 1993). The physical components that make up the system network are vital to the flow of this materiel. These physical components can be categorized as nodes, modes, and routes.

The theater distribution system functions along physical lines of communication (LOC), which account for the theater's transportation assets, geography, and area of operations. Nodes are locations within the LOC where logistical support is originated, processed for onward movement, transferred to another transport node, or terminated (FM 55-10, 1999). Modes are the transportation assets that move cargo and personnel between nodes. Figure 1.2 is a geographic display of a theater distribution network and its LOC.

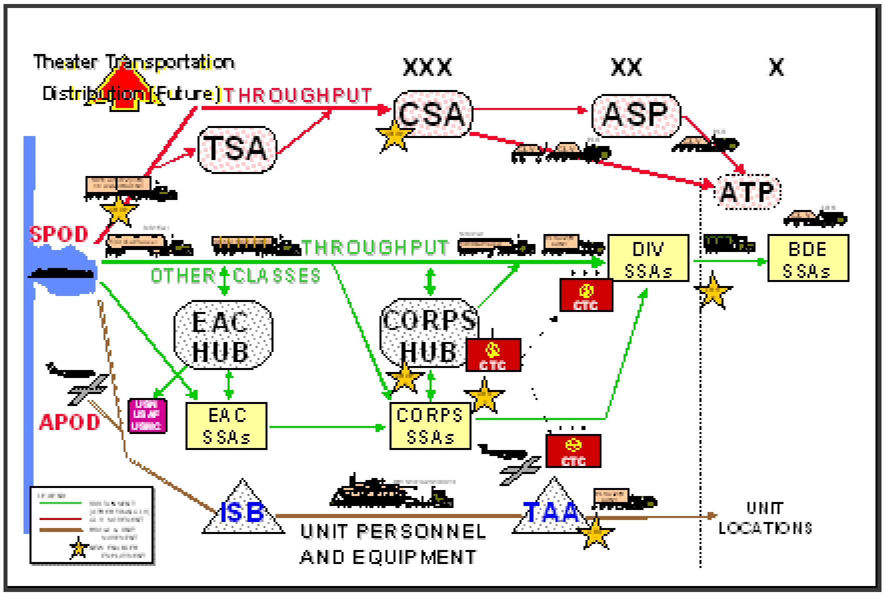

Figure 1.2 Theater Distribution Geographic Network (FM 55-10)

The nodes in a physical theater distribution network consist of the following types: airports, river terminals, sea ports, instream off-load sites, railheads, staging areas (SAs), supply support activities (SSAs), trailer transportation points (TTPs), storage activities, and ammunition supply points (ASPs) (FM 55-10, 1999). The nodes depicted in Figure 1.2 are a representation of nodes when moving materiel and ammunition. These nodes are described in further detail below. They are used when presenting the theater distribution network for the research problem specification. Nodes similar to the nodes discussed below exist for other commodity types such as personnel and petroleum.

- APODs (aerial ports of debarkation) – APODs are airfields used for the sustained air movement of personnel and materiel or to serve as an authorized port for entrance into or departure from the theater of operations.

- SPODs (seaport of debarkation) – SPODs are seaports that are responsible for the sustained movement of equipment and materiel into and out of the theater of operations.

- SSA (supply support activity) – A location that services, stores, and distributes incoming and outgoing materiel. SSAs are located at the theater (TSA), Corps (CSA), Division (DSA), and Brigade (BSA) levels.

- ASP (ammunition supply point) – A location that services, stores, and distributes ammunition.

- Hub – A distribution terminal where materiel may be shipped to SSAs or other nodes as required. Also acts as a receiver, temporary storage, distribution, documentation, and redirect center.

- ISB (Intermediate Staging Base) - An intermediate staging base provides a logistics support base for deploying units in transit to a combat theater or other mission.

- TAA (Tactical Assembly Area) - An area that is generally out of the reach of light artillery and the location where units make final preparations (pre-combat checks and inspections) and rest, prior to moving into combat.

There are many different types of transportation modes between geographic nodes in a theater distribution network. The classes of modes are air, ground, rail, water, and pipeline. Each class has specific characteristics that determine its use within a distribution network.

Air transportation assets provide the most expeditious form of delivery. This method of transportation is a means to quickly move mission essential traffic into theater. It provides the capability to move cargo and personnel to areas where other forms of transportation are restricted by terrain. Airlift is limited by climate and traffic ability of takeoff and landing areas. It also tends to have high ton-miles operating costs (FM 55-10, 1999). Two types of airlift used in theater are Army rotary wing aircraft and Air Force fixed wing aircraft. The Air Force aircraft provide delivery of logistics from theater entry point APODs to Corps areas. Army rotary wing aircraft supplement aerial delivery of high priority items further into theater.

Ground transport modes provide a means to move logistics further into theater where Air Force aircraft are limited. This is the primary mode for distribution operations below Corps level. It is the most flexible mode of transportation and provides traffic ability in practically all weather conditions. It has the capability to transport nearly any commodity with a variety of specialized equipment for both on and off road movement. It is limited by route interferences and obstacles created by weather, terrain or enemy action. It is costly in terms of ton per mile output versus expenditure of manpower and equipment (FM 55-10, 1999) relative to Air Force airlift assets.

Water transportation is primarily used as a means to deliver logistics over oceans and through inland waterways. It can move any commodity under any weather condition. It is the most economical long distance carrier. Unfortunately, water transport modes are relatively slow and are constrained by waterways, facilities, and channels.

Rail provides an inland mode for sustained flow of large quantities of supplies over long distances. It can move any commodity under any weather condition. It provides the most economical continuous line-haul operations and greatest sustained ton per mile capability. However, rail flexibility is limited by fixed routes and by the host nation rail system infrastructure (FM 55-10, 1999). The Army does have one deployable transportation railway battalion, which is capable of building 160 - 200 km of rail for the purposes of setting up a rail network (FM 100-10-1, 1999).

The pipeline provides an economical means to transport bulk liquids. Its flexibility is limited by immobile facilities and is vulnerable to sabotage and enemy action (FM 55-10, 1999).

All transportation modes are considered when developing a theater distribution plan. Some modes are better suited than others in supporting the commanders' logistical operations.

Lines of communication for a theater distribution plan also include route specification. Routes are interdependent with the nodes and modes within a theater distribution system. Depending on the mode of transportation, a route between two nodes may differ. For example, if there exists a large body of water between two nodes, then ground transport is obstructed and must go around, whereas, the route for air and water transport modes is a straight line. Another example is the restriction of no fly zones for aircraft, whereas, ground transport may route their vehicles through that geographical area.

Movement control planners develop the routes used by the various transportation modes within the theater. In their plans, they consider traffic ability, security, geography, and tactical operations. Traffic ability often depends on the existence of established road and rail networks, bridges, tunnels, and other critical points. Geography includes proximity of routes to established facilities and operational locations.

In summary, the physical theater distribution system is a network comprised of nodes, routes, and modes of transport between nodes. The primary nodes consist of PODs, SSAs, and Hubs. The modes of travel are air, ground, water, rail, and pipeline, where each are characterized by unique capabilities and limitations. Movement control planners define the routes that transportation modes use for delivering logistics between nodes.

1.1.3 Theater Distribution Model Requirements

This section provides modeling requirements for the theater distribution model. The requirements were extracted from Joint, Army, and Air Force doctrinal manuals, logistics journals, a theater distribution management system feasibility study (HQ USAF/ILA, 2002), subject matter experts, conferences, and other relevant publications. The Joint publications provide an overview of the command and control, roles and responsibilities, tenets, fundamentals, and operational considerations of theater distribution. The Army manuals provide a more explicit reference on theater distribution planning and execution. The Air Force manuals provide some roles and responsibilities, but they do not provide explicit modeling requirements. The Air Force does, however, have a document titled "Theater Distribution Management System Requirements and Feasibility" (HQ USAF/ILA, 2002), which outlines requirements for a comprehensive theater distribution model.

The requirements listed in this section are not exhaustive. Instead, they are general requirements used for modeling purposes. Many of the real world modeling requirements not specifically listed can be aggregated into the general requirements. For example, a general requirement for the theater distribution model is loading and/or unloading times at a depot and/or customer. In the theater distribution model, this general requirement is represented as a period of time. However, there are numerous tasks (other than loading or unloading) that occur during the time period when vehicles are docked at a location.

The theater distribution modeling requirements are grouped by nodes, modes, and routes, as previously described for a theater distribution physical network. The nodes are

categorized as depots, hubs, and customers. The depots are locations that distribute cargo. They also have vehicles assigned to the location, for the purpose of distributing cargo. Hubs are locations that receive, store and distribute logistics. The customers are locations that only receive logistics. For the purpose of this model, they do not have vehicles assigned to them. Depots are SPODs and APODs. Hubs are transshipment points. Customers are TSAs, CSAs, DSAs, or BSAs.

Modes, or vehicles, are means of transporting cargo and personnel between nodes. The requirements are general enough to account for the air, ground, water, and rail modes. The pipeline mode requirements are somewhat different and are not considered in this research.

Given below is a brief explanation of each of the modeling requirements grouped by mode, node, and route.

1. Vehicles
    a. The vehicles of various types are referred to as multiple nonhomogeneous vehicles (MV$\overline{\text{H}}$)
    b. The vehicles are owned by US services (Army, Navy, AF), US contracted (civilian), and/or host nation
    c. The modes of vehicles are rail, air, ground, water
    d. Vehicles have fixed and/or variable costs depending on the owners
    e. Vehicles are constrained by capacities (by weight and/or volume)
    f. Vehicles are constrained by length (distance) of travel per route (**RL**)
    g. Vehicles have load and unload times at customer and depot locations
    h. Vehicles may be constrained by the type of cargo they can deliver
    i. Vehicles may be limited to certain routes they can travel
    j. Vehicles may be allowed multiple trips (**MT**) or single trips (**ST**) per planning horizon (time period)
    k. Vehicles may be constrained by crew rest times between trips
    l. Vehicles may perform direct delivery from outside the theater
    m. Vehicles may refuel at hub and/or customer locations

2. Depots
    a. There exists multiple depots (**MD**) or a single depot (**SD**) in each theater
    b. Vehicles are assigned to depots by type and quantity
    c. Depots are limited by throughput capacity via materiel handling constraints and other resources (**wMOG**).
    d. Depots may have multiple time windows (**MTW**), where certain vehicle types can only depart/arrive within that time period

3. Hubs (**H**)
    a. Hubs receive, store, and distribute logistics
    b. Hubs have storage constraints
    c. Hubs have parking maximum on the ground (**pMOG**) constraints
    d. Hubs are limited by throughput capacity via materiel handling constraints and other resources (**wMOG**).
    e. Hubs may have multiple time windows (**MTW**), where certain vehicle types can only deliver within that time period
    f. Hubs have vehicle assets assigned to them to distribute logistics
    g. Hubs distribute logistics after they are received and processed.

4. Customers
    a. Customers have specified demands
    b. Customers may have delivery time window (**TW**) or multiple time window (**MTW**) constraints
    c. Customers may have **m**aximum **o**n the **g**round (**MOG**) constraints
    d. Customers may have time definite delivery (**TDD**) requirements
    e. Customers may have no earlier than delivery times (**ETDD**)
    f. Customers are prioritized by the commander's operational plan
    g. Customers may require multiple services (**MS**) or a single service (**SS**) per planning horizon

5. Routes (LOCs)
    a. Movement control planners stipulate routes
    b. Routes are the paths between nodes
    c. Routes have differing travel times for different vehicle types
    d. Vehicle types may have unique routes
    e. Routes may consist of a direct delivery from outside the theater

Theater distribution model requirements also include the intent or purpose of the modeling effort. Section 1.2 provides an introduction to why a theater distribution model interests the military community and its purpose for use. Basically, the model's purpose is to increase the efficiency and effectiveness of theater distribution by

prescribing vehicle routes and schedules that minimize cost and maximize customer satisfaction. The model stakeholders intend to ensure delivery of the "right things" to the "right place" at the "right time" (JP 4-01.4, 2000). In terms of the model, efficiency is the ability to provide cargo and personnel at minimal cost. Effectiveness is the ability to provide cargo and personnel to the customer that meets their demand and time requirements. Efficiency and effectiveness may be defined in numerous ways and are specifically defined later in the model's objective function.

1.2 Theater Distribution Problem Modeling Interest

The initiative to build a model that integrates multi-modal transportation assets to more efficiently and effectively plan and execute logistics support within a theater has become a priority among the services. This section provides some insight as to why the services are interested in a theater distribution model.

1.2.1 General Logistics Trend

In a keynote address, the Honorable Paul G. Kaminski, Under Secretary of Defense for Acquisition and Technology, discussed the "Revolution in Defense Logistics" (Kaminski, 1995). He stated that logistics has finally become a growing concern among warfighters. They have come to realize that the role of logistics has grown more crucial as modern warfare increases in technological sophistication, cost, speed, and complexity. Commanders must have logistical visibility in order to properly execute their strategic and tactical operations.

Providing necessary logistics comes with a cost, and given the current Department of Defense budget, every dollar wasted on inefficient logistical operations is a dollar

taken from building, modernizing, or maintaining our war fighting capability (Kaminski, 1995). Today, with no clear capable enemy in the post cold war environment, there has been greater pressure on the defense budget. Smaller more responsive logistics approaches that require less investment and money to operate are being aggressively pursued (Schrady, 1999).

To become more efficient, it is paramount the military move away from the just-in-case inventory approach, which is a costly stockpile of inventory (Schrady, 1990). This requires a move from a layered inventory system to a process that provides rapid and reliable transportation that moves inventory to the customer as required.

The military has confronted logistical issues throughout history. Most recently, the Gulf War in 1991 and Operation Allied Force in 1999 provided feedback on logistical shortfalls.

1.2.2 Gulf War and Operation Allied Force Logistics Lessons

The Gulf War provided some logistical lessons that prompted requirements for logistical process improvement. General Schwarzkopf understood the necessity for appropriate logistical support and required (in theater) a thirty to sixty day supply of most sustainment materiels. As a result, these huge stockpiles of supply took months to accumulate and left a huge and vulnerable "footprint".

It was noted that "probably the worst decision of Desert Shield/Storm was the decision to stock 60 days of supply and ammo in-country. It drove up force structure, it cost the Army lots of money and time, and over 90% was backhauled" (Foss, 1994). This resulted in costs that may have been avoided with a more efficient logistics system.

Operation Allied Force also provided logistical lessons. The Air Force documented two logistic shortfalls: determining sustainment requirements and theater distribution planning. Units were already bedding down at bare base locations and no theater distribution plan was in place to provide for their sustainment needs or for those that would deploy to the theater in the near future (Brooks, 2000).

As a result, the Air Force initiated the development of an automated theater distribution system, which develops the Air Force's portion of the joint theater distribution plan. This automated Theater Distribution Management System (TDMS), when complete, will enable logistics planners to react to rapidly developing contingency operations common to the expeditionary Air Force (Brooks, 2000). The Air Force hopes this model will scale to meet the other services theater distribution modeling requirements.

1.2.3 Service Interests in a Theater Distribution Model

As mentioned in previous sections, a more streamlined logistics process is beneficial for all the military services. Each service has an initiative that makes their logistics process more efficient and effective. Each of these plans support the philosophy of "Focused Logistics" in the Chairman of the Joint Chiefs of Staff's Joint Vision 2010 document. Joint Vision 2010 states, "focused logistics will be the fusion of information, logistics, and transportation technologies to provide rapid crisis response, to track and shift assets even while enroute, and to deliver tailored logistics packages and sustainment directly at the strategic, operational, and tactical levels" (Shalikashvili, 1996).

Supporting the Focused Logistics concept is the Army's concept of "Velocity Management," the Air Force's concept of "Lean Logistics," and the Navy's concept of "Seabased Logistics." The goal of the Army's program is to make Army logistics as fast and efficient as a Fortune 500 company (Barnes, 1996). Velocity Management focuses on simplifying logistical processes, substituting velocity for mass and implementing improvements to the system. The Air Force's program plans on moving from a supply (inventory) based to a transportation based logistics system (Schrady, 1999). The Lean Logistics approach is a consolidation of wholesale inventories, a drastic reduction of base level inventory, and a new focus on customer mission requirements through a high velocity transportation infrastructure. The Navy's program objective is to reduce the "footprint" ashore by operating from a base of ships at sea, thus minimizing vulnerability. Reducing shore-based logistics reduces the shore-based manpower requirements, thus resulting in lighter, more agile tactical forces operating on land (Schrady, 1999).

Although each service has initiatives to improve their logistics systems, none have an automated theater distribution model, and the need is widely recognized (Brooks, 2000). Consequently, the Commanders in Chiefs (CINCs) have classified the idea of an automated TDMS as a category one requirement. This requirement is specified as requirements 18, 25, 33, and 38 of the CINC's 57 category one requirements (Brooks, 2000).

A TDMS will most likely be utilized at the CINCs' logistics staff and service component levels. The CINCs' logistics staff, service component logistics staff, and service component operational units, which are linked together to operate and perform day to day distribution functions, are responsible for execution of the theater distribution

mission (JP 4-01.4, 2000). An automated TDMS should have the capability to integrate theater logistical requirements with multi-modal theater distribution assets.

1.2.4 Theater Distribution Model Capability

The general capabilities of a theater distribution model were outlined in a position paper on the theater distribution management system initiative (Brooks, 2000). Although the initiative was primarily to meet the logistical concerns of the Air Force, he did recognize the necessity for this model to incorporate multi-modal transportation assets. He mentioned that TDMS should include theater air, ground and water transportation capabilities to ensure an optimal distribution plan is developed for the theater.

The TDMS capabilities should also include its integration with existing logistics models. These models include the Global Combat Support System (GCSS), the Logistician's Contingency Assessment Tools (LOGCAT), and the Sustainment Planning Tool (SPT). The GCSS is a communications network that provides an integrated combat support infrastructure for all combat support areas, including acquisition, logistics, personnel, finance, and health services (JP 4-01.4, 2000). LOGCAT, which consists of the Survey Tool for Employment Planning (STEP) and Beddown Capability Assessment Tool (BCAT), provide users beddown location information and the capability to support a given force structure (Brooks, 2000). SPT predicts and forecasts sustainment requirements by supply class for each of the theater beddown locations (Brooks, 2000).

Once a planner receives SPT output, he/she performs a "stubby pencil drill" of creating a theater distribution plan. Given today's automated modeling capability, this kind of planning should be automated. Therefore, an automated application that

combines sustainment requirements and theater transportation capabilities to create an optimal theater distribution network is highly desirable (Brooks, 2000).

There are a number of modeling requirements specified by Brooks. He mentioned the model's ability to cover immediate sustainment and time-definite delivery of assets. It should build on the information input into the SPT and additional information such as: beddown locations, type of aircraft or mission, number of aircraft, projected sortie rates, expenditure rates, aerial ports of embarkation (APOEs), APODs, SPODs, surface transportation capabilities, retrograde cargo, in-theater repair/forward stockage facilities, host nation support/participation, and base population to create a dynamic theater distribution plan (Brooks, 2000).

1.2.5 Current Theater Models

As previously mentioned, there are no models that perform the functions specified for theater distribution. However, there are models available that are closely related to a model of this type. Some currently in use are the Scenario Unrestricted Mobility Model for Intratheater Simulation (SUMMITS), Enhanced Logistics Intratheater Support Tool (ELIST), Deployment Analysis Network Tool Extended (DANTE), and Tactical & Seabased Logistics Distribution Systems (T.LoaDS & C.LoaDS) models.

SUMMITS is a simulation model used by the Air Force to measure performance of transportation assets that transport troops and equipment from airfields, seaports, and pre-positioned sites in theater to destination airbases, staging areas, and tactical assembly areas. It considers delivery dates, payloads, rates of movement, loading and unloading times, and the available transportation assets and network capabilities. It examines every

feasible path from origin to destination and selects the fastest path through the network (GAO, 1998). This model was used to support the Air Force's Intratheater Lift Analysis (ILA).

ELIST is a simulation model used by USTRANSCOM's Military Traffic Management Command (MTMC). The model simulates a "fort to port" and Joint Reception, Staging, Onward Movement, and Integration (JRSO&I) process by flowing a TPFDD over a theater transportation infrastructure. The purpose of the model is to help the theater distribution infrastructure planning process. ELIST helps determine closure rates of personnel and materiel to their theater locations, potential bottlenecks, required assets, port throughput/clearance, and other issues and constraints.

DANTE is an analysis tool for studying large-scale deployment scenarios. It represents a deployment from bases in the United States, from forward-based locations and from pre-positioned stocks as a time-phased network flow. The objective is to minimize the time to close the deploying force on the staging area (Hodgson et al., 2001). The model is currently used by the MTMC Transportation Engineering Agency.

T.LoaDS is a simulation application for assessing current or future tactical or sea-based distribution systems. In its current state of development, it is an analytical model for assessing the pros and cons of new doctrine, distribution techniques, organizational structures, and equipment concepts (Hamber, 2001). T.LoaDS is currently undergoing final development and is sponsored by the Office of Naval Research.

These models achieve their intended purposes, but they do not provide the information as prescribed earlier for a real time automated theater distribution sustainment-planning tool.

1.3 Theater Distribution Problem Modeling Techniques

The theater distribution problem in this research is formulated and solved by a synthesis of techniques. Section 1.3.1 explains how the theater distribution problem conforms to and is formulated as a vehicle routing and scheduling problem (VRSP). Section 1.3.2 discusses the use of the tabu search metaheuristic as an approximation algorithm to solve the theater distribution VRSP. Section 1.3.3 introduces the use of group theory to enhance the tabu search metaheuristic. Section 1.3.4 discusses the use of Java™ programming to build and solve the theater distribution VRSP model.

1.3.1 Theater Distribution as a VRSP

When evaluating requirements, it becomes obvious that a theater distribution model can be configured and solved as a VRSP. A VRSP is a network that routes vehicles to customers, schedules the customer services and routes, and adheres to any precedence relations in the network.

Movement control planners who develop the theater distribution movement program perform the following steps (FM 55-10, 1999).

1. Assess the plan
2. Determine the requirements
3. Determine transportation capabilities
4. Balance requirements against capabilities
5. Determine shortfalls, critical points, and recommend solutions for handling the shortfalls
6. Coordinate the program
7. Publish and distribute the program

When movement planners assess the plan, they create a distribution network based on a complete logistics picture that shows locations of the entire logistics infrastructure. They also forecast requirements. Requirements are in the form of objectives, demands, time windows, and other constraints. To determine capabilities, movement planners ascertain the number of transportation assets, location and number of materiel handling equipment and the level of storage available in the transportation network. The most important step in achieving an efficient movement plan is balancing requirements against capabilities. Movement planners are tasked to assign routes and loads to vehicles in order to meet the commander's intent given available resources and constraints. This step can be solved as a VRSP, which will provide the information that movement planners require. The last step for discussion is determining shortfalls. Movement planners desire feedback where the plan does not meet all the requirements. A VRSP, modeled with the appropriate constraints, should provide necessary information for planners to use.

1.3.2 Introduction to the Tabu Search Metaheuristic

A theater distribution plan is neither a precise nor static document. It relies heavily on forecasted demand and there are many unknowns that can easily affect the plan at any moment in time. Therefore, movement planners must be flexible as requirements often change (FM 55-10, 1999).

Vehicle routing and scheduling problems can be solved by both optimization methods and approximation methods. Optimization methods like branch and bound find the optimal solution, but they tend to require tremendous computation time to find the

optimal solution for large problems. Approximation methods do not guarantee an optimal solution, but they do tend to find satisfactory solutions in less time than optimization methods for similar size problems.

Data for a theater distribution problem are very inexact; consequently, an optimal solution may not be applicable to the actual situation. Therefore, an approximation method that quickly provides a satisfactory and robust solution is highly preferred by theater distribution movement planners.

There are many approximation methods used to solve VRSPs. The best of these is tabu search, a metaheuristic developed by Glover (Glover and Laguna, 1997), which has been used quite extensively over the past decade to solve VRSPs. In recent years, tabu search applications have provided the best solutions in the least amount of time for many instances of the VRSP. Tabu search exploits a collection of principles of intelligent problem solving including forms of adaptive memory and responsive exploration (Glover and Laguna, 1997). These principles, when used effectively, provide a framework for solving the theater distribution problem.

1.3.3 Group Theory Application

Part of the tabu search methodology is the use of a move neighborhood. A **move** is defined as an operation on an incumbent solution that changes it to a new solution. A **move neighborhood** is defined as the set of solutions adjacent to the incumbent, i.e. reachable in a single move. Tracking moves and move neighborhoods can be cumbersome and computationally burdensome. Until recently, no generalized methods existed for overcoming such difficulties.

In a groundbreaking dissertation, Colletti (1999) pioneered the association of group theory, a subset of abstract algebra, with metaheuristics. Prior to Colletti's work, Gomory (1958, 1963) successfully applied group theory to develop the cutting plane method for integer programs. Colletti identified the symmetric group on n-letters, S_n, as a natural setting for solving permutation type problems with metaheuristic techniques (Colletti, 1999).

From Colletti's work, a new methodology termed Group Theoretic Tabu Search (GTTS) is currently under research and development. GTTS is the utilization of group theoretic constructs with tabu search to solve combinatorial problems. Wiley (2001) was the first to apply GTTS.

For this research, S_n is exploited as a means to structure the moves and move neighborhoods in the tabu search metaheuristic and to provide efficient and effective solution searches. In Section 3.2, the salient aspects of S_n to this research are presented.

1.3.4 Introduction to the Java™ Software Programming Language

The military requires a solution method, which is flexible and powerful enough to solve real world theater distribution problems. The Java™ programming language and its object-orientation provides a heretofore-unrealized generalization of software architectures that significantly assists in providing such a solution (Harder, 2000).

Java™ is a relatively new programming language that has found its way into the operations research community as a versatile means to code algorithms. Java™ adapts features from other languages, runs on any machine, provides convenient access to the

internet, and is easy enough to allow novice programmers to produce programs with fairly sophisticated user interfaces (Arnow and Weiss, 2000).

Previous work in the area of Java™ programming for tabu search algorithms was performed by Harder (2000) and Wiley (2001). Harder developed a general tabu search engine that frees analysts from writing controlling code, so they may concentrate on defining the specifics of the tabu search. The tabu search engine and required tabu search classes are displayed in Figure 1.3.

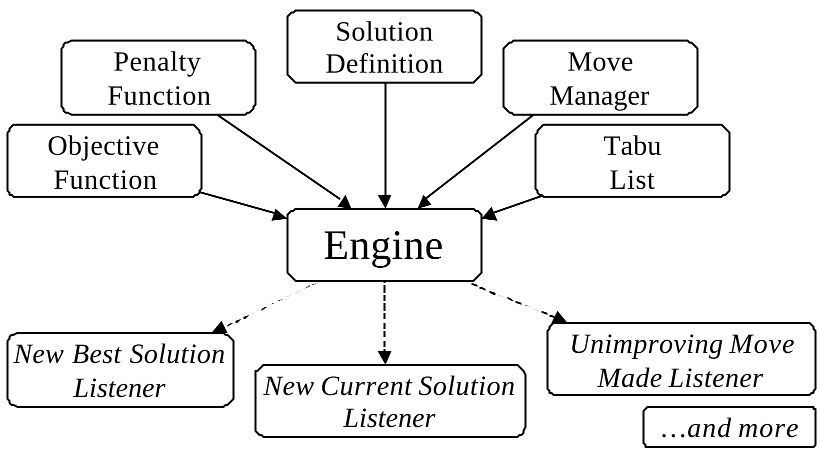

Figure 1.3 Tabu Search Engine Architecture (Harder, 2000)

Harder's code provides for the engine and class interfaces of the objective function, penalty function, solution definition, move manager, and tabu list classes. The analyst is required to develop the classes required for the interface.

Harder's tabu search engine and interfaces are used in the code that solves the theater distribution VRSP. Others have successfully used Harder's code in their research, which includes Cullenbine (2000), Hall (2000), Calhoun (2000), Kinney (2000), Wiley (2001), and Brown (2001).

Wiley's contribution to Java^{TM} programming and tabu search is the development of a Java^{TM} class library (Wiley, 2000) that supports the application of group theory to tabu search. The library consists of a Group class and a Symmetric Group class. These classes allow the user to represent the tabu search moves and solutions as symmetric groups on n-letters. Within the classes, Wiley defines data members that represent S_n objects, provides constructor methods for instantiating S_n objects, and supplies other support methods for manipulating S_n objects. These predefined classes save the user time and effort when developing the tabu search algorithm.

1.4 Theater Distribution Vehicle Routing and Scheduling Problem Definition

Section 1.4 transforms the theater distribution problem into a theater distribution vehicle routing and scheduling problem (TDVRSP). Section 1.4.1 defines the characteristics of a generalized vehicle routing problem (GVRP) and additional characteristics required by the TDVRSP. Section 1.4.2 presents a hierarchy of the TDVRSP. Section 1.4.3 provides TDVRSP hierarchy cases and how they relate to actual military application.

1.4.1 TDVRSP Model Characteristics

Carlton (1995) proposed a three-tiered representation of a GVRP displayed in Figure 1.4 below. The first tier is based on the traveling salesman problem (TSP). The second tier is based on the vehicle routing problem (VRP), a capacity constrained TSP. The third tier incorporates precedence to the VRP characterized by the pickup and delivery problem (PDP). Within each tier, the number and type of vehicles, number of depots, route restrictions and time restrictions characterize the problem.

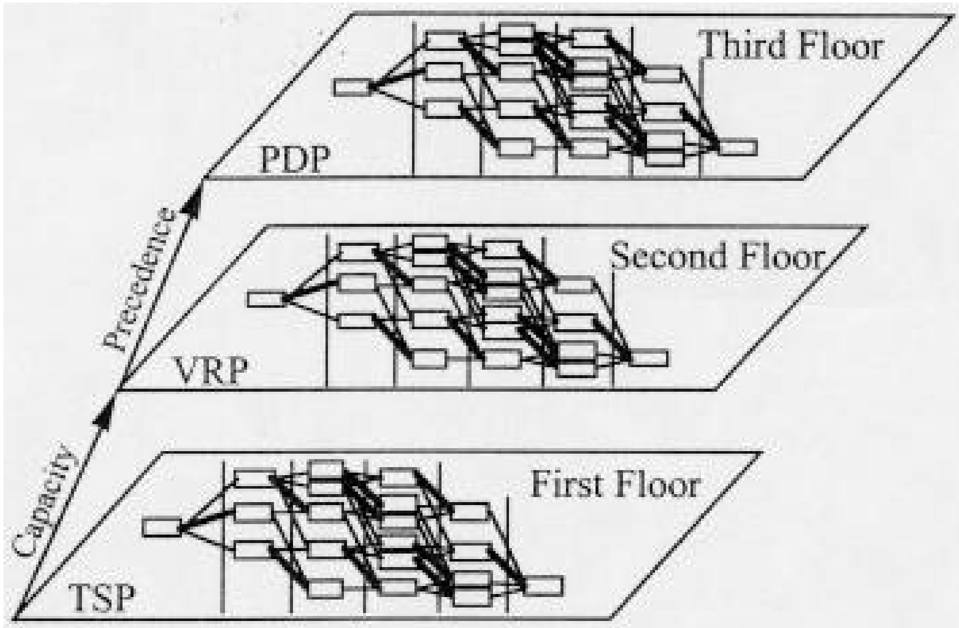

Figure 1.4 Carlton's GVRP Hierarchy (Carlton, 1995)

The second floor, Figure 1.5, is of most interest to this research. Carlton's GVRP second floor includes the following cases and their possible combinations:

1. SV: Single vehicle case
2. MVH: Multiple homogeneous vehicles
3. $MV\overline{H}$: Multiple nonhomogeneous vehicles
4. SD: Single depot
5. MD: Multiple depot

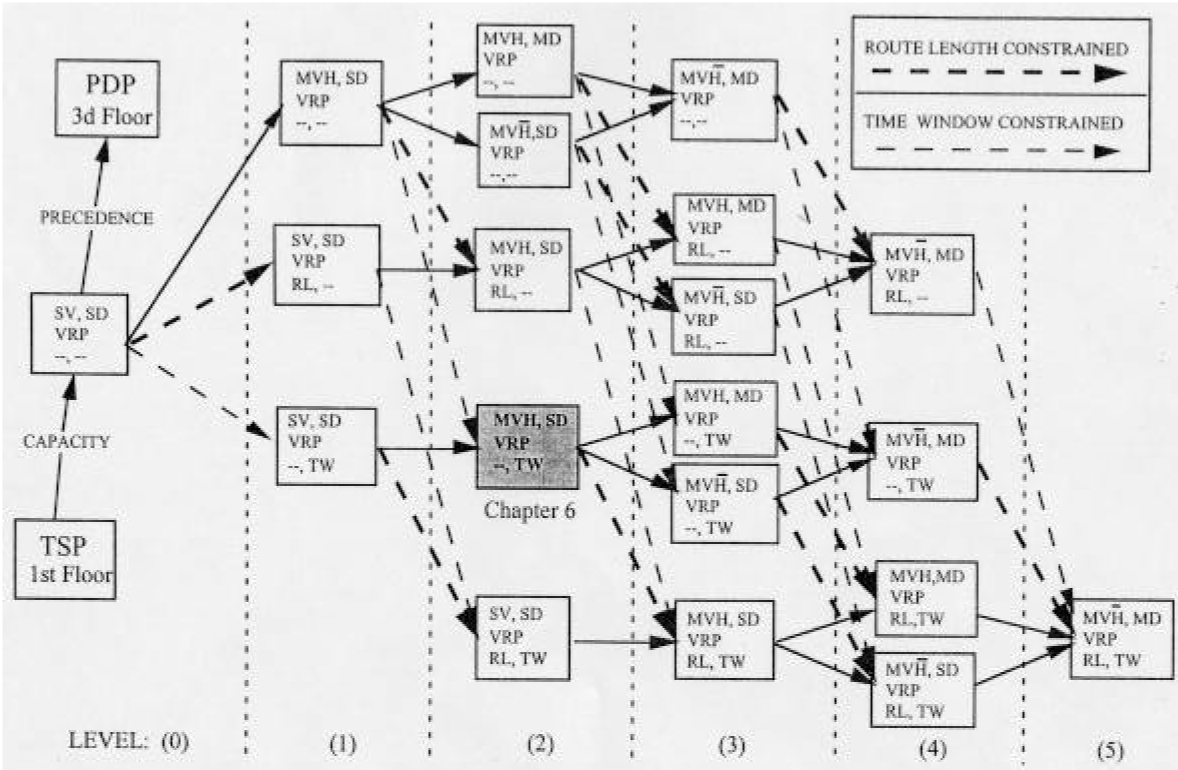

Figure 1.5 Carlton's GVRP Hierarchy Second Floor (Carlton, 1995)

A single vehicle case is a VRP where only one vehicle services all the customers. The multiple homogeneous vehicle case is where more than one vehicle of the same type services all the customers. The multiple nonhomogeneous vehicle case is where differing types of vehicles service the customers. The single depot case is where all the vehicles originate and return to the same depot. Vehicles are assigned to more than one depot in the multiple depot case.

The VRP is further characterized by route length (RL) and time window (TW) constraints. Route lengths restrict the amount of time or distance a vehicle can travel. A time window constraint is a block of time that vehicles may service a customer. Both the RL and TW constraints create more instances of the VRP. There are a total of fifteen

VRP instances in the second tier. The most general problem in Carlton's taxonomy is the MD, MV$\overline{\text{H}}$, VRP, RL, TW, which is described as the multiple depot multiple nonhomogeneous vehicle, vehicle routing problem with route length and time window constraints.

The theater distribution VRSP hierarchy augments Carlton's hierarchy with four more model characteristics. The additional characteristics include:

1. ST: Single trip (per vehicle) versus MT: Multiple trips (per vehicle)
2. SS: Single service (per customer) versus MS: Multiple service (per customer)
3. H: Hubs (transshipment nodes)
4. SC: Single commodity versus MC: Multiple commodity

The single trip case is where vehicles may only make one trip per planning period. The multiple trips case allows vehicles to make more than one trip per planning period. The number of trips allowed per vehicle are limited for each type. The single service case allows only one service per customer per planning period. The multiple services case allows more than one service per customer. The hub is a node that acts as both a depot and a customer. The hub has vehicles assigned to the location whose purpose is to service customers. Vehicles from a depot or higher-level hub also service the hub. The single commodity does not differentiate between commodity types and serves the case where there is only one commodity. The multiple commodity case explicitly accounts for more than one commodity.

Section 1.4.2 combines the added model characteristics into a TDVRSP hierarchy. The hierarchy consists of sixteen possible combinations. The most general problem is the MD, MV$\overline{\text{H}}$, VRP, RL, TW, MT, MS, H, MC. Each theater distribution VRSP hierarchy instance is an extension of the most general instance of Carlton's second floor GVRP hierarchy.

1.4.2 Theater Distribution Model Hierarchy

Figure 1.6 is a theater distribution model hierarchy. The hierarchy starts where Carlton's hierarchy ended, i.e., at the MV$\overline{\text{H}}$, MD, VRP, RL, TW, with the added dimensions ST, SS, and SC. These are typically standard assumptions in the VRPTW as described by Carlton (1995). The hub (H) is not part of the starting conditions

The hierarchy displayed in Figure 1.6 below presents 16 possible combinations for the TDVRSP. The dimensions are ordered from left to right based on the complexity they add to the problem. The first column is the basic VRPTW with standard assumptions. The second column includes multiple trips per vehicle. The third column includes multiple services per customer. The fourth column includes hubs in the network. The last column is the multiple commodity dimension.

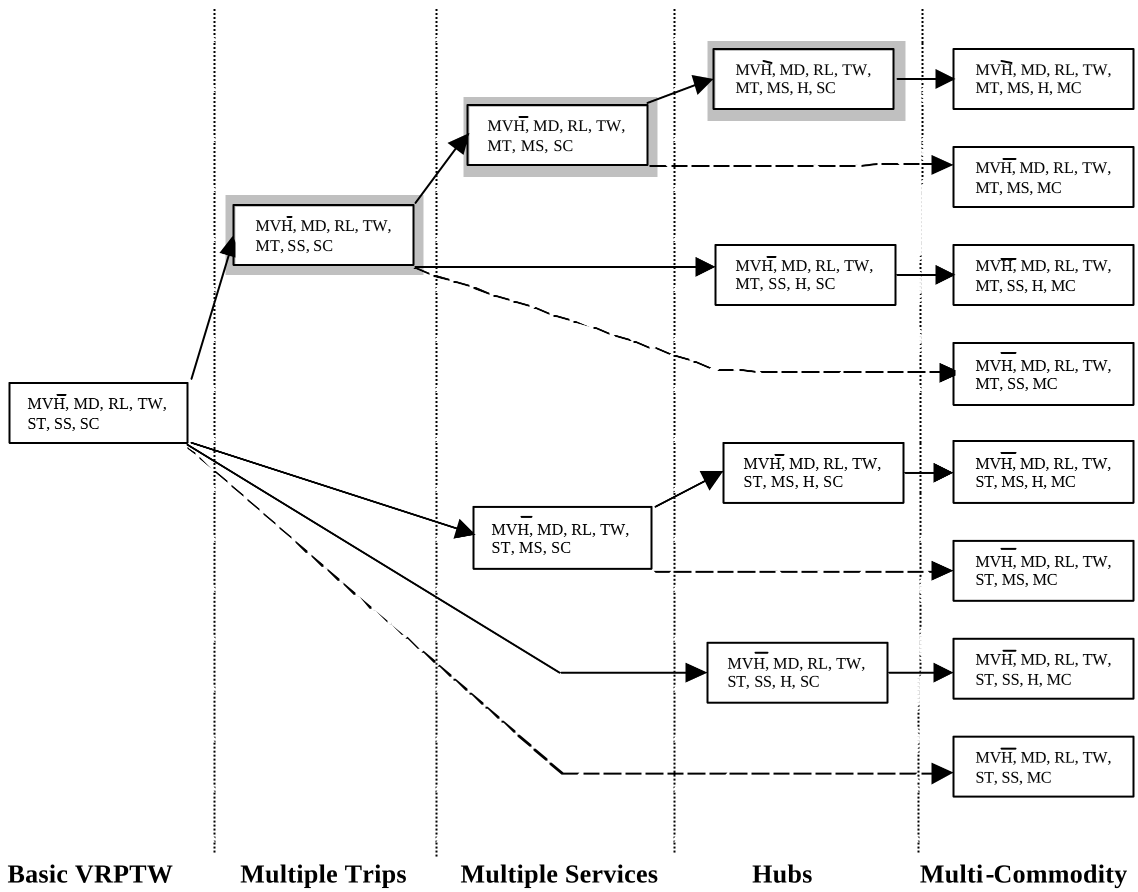

Figure 1.6 TDVRSP Hierarchy

1.4.3 TDVRSP Hierarchy Cases And Military Utilization

The TDVRSP hierarchy cases are applied to specific military distribution applications within the theater. This section presents three generalized hierarchy cases for application. The hierarchy cases discussed are the multiple vehicle trips without multiple service to customers and no hubs, multiple vehicle trips with multiple services to customers and no hubs, and multiple vehicle trips with multiple services to customers and hubs. For the purpose of this research, each case is *limited to the single commodity*

dimension. The multiple commodity dimension can be included in similar applications, and is left for further research. Each case is generalized to include multiple depots or single depots and homogeneous or nonhomogeneous vehicles. Route length restrictions and time window constraints are also incorporated in these applications.

The first application incorporates the generalized multiple vehicle trip case without multiple services to customers and no hubs. The multiple trip case is the MV$\overline{\text{H}}$, RL, TW, MT, SS, SC, VRSP. A small example of an associated distribution network is presented in Figure 1.7. In this example, there exist two nonhomogeneous vehicles. One vehicle and its two routes are represented by solid arcs, whereas, the other vehicle and its route is represented by dashed arcs

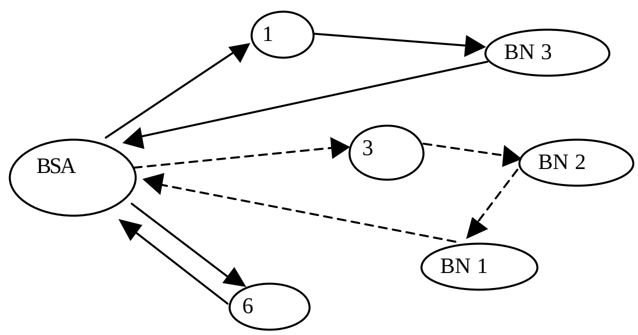

Figure 1.7 MV$\overline{\text{H}}$, SD, RL, TW, MT, SS, SC, VRSP Network

This TDVRSP instance can be applied to cargo distribution within a command's area of interest. A brigade's distribution network, which is only concerned with delivering cargo from the BSA to customers within the brigade area or to the battalion (BN) supply areas, is an example of this case. Distribution most likely involves sustainment operations, where a customer's demand is satisfied by a single visit within a

specified time period. Multiple vehicle trips may be performed to satisfy all customer demands within the time period.

The second military application is based on the multiple vehicle trips with multiple services to customers and no hubs. The multiple vehicle trips and multiple services to customers is the MV$\overline{\text{H}}$, RL, TW, MT, MS, SC, VRSP case. A small example of an associated distribution network is displayed in Figure 1.8.

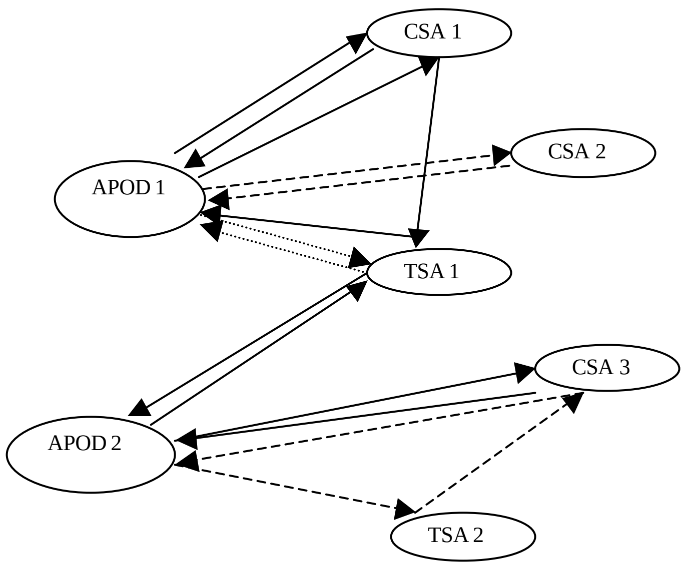

Figure 1.8 MV$\overline{\text{H}}$, RL, TW, MT, MS, SC, VRSP Network

This TDVRSP instance can be applied to the Air Force theater distribution problem, where aircraft deliver cargo from APODs to austere airfields within the theater and corps areas. This application involves delivering cargo over an extended time period where multiple vehicle trips are performed and customers are serviced multiple times within the time period. This instance supports a force projection operation.

In Figure 1.8, the depots are represented as APOD 1 and APOD 2. The customers are CSA 1, CSA 2, CSA 3, TSA 1, and TSA 2. In this example, CSA 1, TSA 1, and CSA

3 are serviced multiple times. Nonhomogeneous vehicles perform multiple trips. At APOD 1, there are three vehicles. The solid arcs represent one vehicle and its two tours. The dashed arcs represent a vehicle and its single tour. The dotted arcs represent the third vehicle and its tour. APOD 2 has two nonhomogeneous vehicles that perform a total of three tours.

The third military application is based on the multiple vehicle trips with multiple services to customers and hubs, MV$\overline{\text{H}}$, RL, TW, MT, MS, H, SC, VRSP. A small example of an associated distribution network is displayed in Figure 1.9.

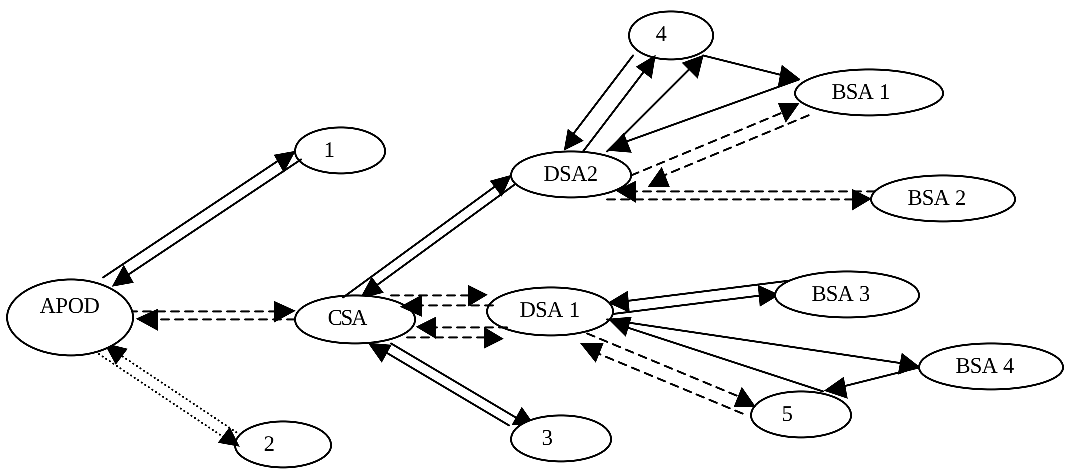

Figure 1.9 MV$\overline{\text{H}}$, RL, TW, MT, MS, H, SC, VRSP Network

This TDVRSP instance can be applied to the theater level distribution operation. At the theater level, single or multiple depots are used, where multiple nonhomogeneous vehicles make multiple trips and provide multiple services to customers within the entire theater. Certain customers act as hubs, where cargo is received, stored and distributed to its customers. Hubs/depots and their customers are termed *tiers* in this research.

In this example, the APOD represents a single depot that distributes cargo to its three customers. The CSA is a hub that receives cargo from the depot and distributes it to

its three customers. The DSAs also act as hubs, where cargo is received and distributed. Similar to the other examples, the nonhomogeneous vehicles and their tours are represented by the solid, dashed, and dotted arcs.

A review of past literature fails to reveal any work performed with multiple vehicle trips and multiple services to customers with or without hubs within an extended time period. Therefore, this research yields the first methodology that provides a means to solve these types of problems.

1.5 Research Objectives

The primary objective is to develop a robust, efficient, effective and flexible generalized theater distribution model that prescribes the routing and scheduling of multi-modal theater transportation assets in order to provide economically efficient time definite delivery of cargo to customers. In doing so, advances are provided in the field of group theoretic tabu search and its application to difficult combinatorial optimization problems, e.g., the multiple trips multiple services vehicle routing and scheduling problem with hubs and other defining constraints.

To accomplish the primary objective, there are a number of supporting objectives. The first supporting objective is to develop a theater distribution vehicle routing and scheduling problem given all its problem characteristics. These characteristics include those from the GVRP plus multiple trips, multiple services, and hubs. This necessitates the collection of theater distribution modeling requirements. Once requirements are consolidated, a TDVRSP hierarchy is developed to characterize the problem instances. The TDVRSP is then formatted to the symmetric group on n-letters and solved by group theoretic tabu search. The end result is a theater distribution vehicle routing and

scheduling problem hierarchy with supporting methods using the symmetric group on n-letters format.

Since benchmark problems do not exist for the TDVRSP, the second objective is to develop a set of problem instances that adequately tests the robustness of the TDVRSP model. The set of benchmark problems requires a methodical design that effectively varies the data to account for possible TDVRSP cases.

The third supporting objective is to create an algorithm that efficiently and effectively solves cases of the TDVRSP. The two primary cases are the multiple trips multiple services TDVRSP with supporting constraints and the multiple trips multiple services with hubs TDVRSP with supporting constraints. The algorithm uses the tabu search philosophy of adaptive memory and intelligent search. Methods of intensification and diversification are incorporated when searching for the solution.

The fourth supporting objective is to utilize group theory and the symmetric group on n-letters to structure the tabu search moves and move neighborhoods. Research is performed on the application of conjugacy classes, cyclic form structures, orbital planes and orbits to partition the solution space and to avoid cycling.

The fifth supporting objective is to code the TDVRSP algorithm with the Java™ software programming language. The software conforms to the architecture of Harder (2000). The software program is then used to determine the algorithm's performance and find solutions for the various TDVRSP problems in the benchmark set.

II. Literature Review

The literature for the General Vehicle Routing Problem (GVRP) and its extensions is significant. Textbooks, special journals, and periodicals relating to the GVRP span the past 40 years. During that time, considerable progress has been made in defining, formulating, and solving the GVRP. There are well over 1000 publications on the GVRP in print.

The literature on group theory and its applications is also abundant. Although there exist numerous references for group theory, there are few references on group theory as it applies to metaheuristics. Many of these references come from the work of Colletti and Barnes. Group theory is introduced in Chapter 3.

This chapter does not provide a complete discussion of all the references on the GVRP. However, it does present an ample review of the GVRP and its instances that are relevant to the research at hand. Section 2.1 provides literature pertinent to the GVRP and basic definitions. Section 2.2 is partitioned into three subsections. The first subsection reviews literature pertinent to the routing instances of the GVRP. The second subsection covers literature pertinent to the scheduling instances. The third subsection is a literature review on combined routing and scheduling instances. Each subsection also discusses methods that recent researchers used to solve the GVRP and its extensions.

2.1 GVRP Literature

The GVRP is well studied. One of the first publications to summarize the GVRP is Bodin and others (1983), which presents a comprehensive review of the GVRP and its instances. Bodin and others categorize the GVRP problem into three classes: routing, scheduling, and routing and scheduling. For each class, they present definitions, notation and formulation. They introduce specific instances of each class type. For example, the routing class instances include the Traveling Salesman Problem, Chinese Postman Problem, Single Depot - Multiple Vehicle - Node Routing Problem, and a variety of others. They also provide a discussion of methods used to solve each GVRP problem class.

Bodin (1990) followed up with an article discussing his 20 years of experience in solving practical routing and scheduling problems. He presents three important components of the vehicle routing problem. They are the algorithmic component, computational environment and the role of geographic information. He states that in practice, most vehicle routing problems have the following characteristics: multiple vehicle types, vehicle/location dependencies, depots, time windows, route length restrictions, and an objective function that minimizes single or multiple objectives. Incorporating all these characteristics plus additional constraints make the VRSP computationally complex. Therefore, heuristics are employed to find a solution.

Carlton (1995) provides a comprehensive study of the GVRP. In his dissertation, he presents the GVRP and provides an in-depth review of the literature concerning the

GVRP. He further presents a classification scheme for the GVRP. The bulk of his work supports the application of reactive tabu search to solve the VRPTW.

2.1.1 GVRP Definition

A typical GVRP is a combinatorial optimization problem that minimizes the cost of routing a fleet of vehicles in order to provide a service to a set of customers with demands. The specific instance of this definition determines the class of GVRP. Bodin and others (1983) categorize the GVRP into three classes: a vehicle routing problem (VRP), a vehicle scheduling problem (VSP), and a vehicle routing and scheduling problem (VRSP). A typical vehicle routing problem is spatial and no temporal considerations are imposed on the problem. A vehicle scheduling problem considers both spatial and temporal factors. A vehicle routing and scheduling problem is typically an application that considers both spatial and temporal factors characterized by task precedence and time window constraints (Bodin and others, 1983). This section provides definitions and notation for the GVRP.

The GVRP, also denoted MV$\overline{\text{H}}$, SD, VRP, RL, consists of minimizing the cost of traveling routes by a fleet of nonhomogeneous vehicles with capacity restrictions and route length constraints. The fleet of vehicles provides service from a depot to a set of customers with deterministic demands. Each customer is visited only once where a single vehicle fills its demand. The fleet of vehicles operates out of a single depot, where the vehicle starts and ends its route at the depot. Each vehicle performs at most one tour per time period. A graphical depiction of this instance is given in Figure 2.1. The numbered circles are customers and the directed arcs represent vehicle routes. In this

example, there are three vehicles that leave a single depot, service seven customers, and return to the depot. The total cost of the solution is the cumulative lengths of the three tours.

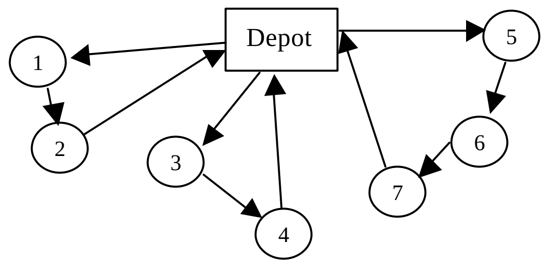

Figure 2.1 Simple GVRP Graphical Example

Bodin and others (1983) provide a formulation of the GVRP. They also include equations for various constraints affecting GVRP instances.

2.2 GVRP Class Instances

The previous section described the GVRP. This section presents specific instances of the GVRP as they apply to this research. Section 2.2.1 provides instances of the vehicle routing problem (VRP) and Section 2.2.2 discusses instances of the vehicle scheduling problem (VSP). Each section provides a discussion of the literature pertinent to that problem instance.

2.2.1 Vehicle Routing Problems

The vehicle routing problem is a class of the GVRP where the route time constraint is omitted (Bodin and others, 1983) and potentially replaced by a route distance constraint. This instance of the GVRP considers only spatial factors. In Desrochers and others (1990), they present a thorough classification scheme for VRP

instances. A few of the instances include: multiple/single depots, types of customer demand, address selection constraints, node or edge routing, homogeneous versus nonhomogeneous vehicle types, commodity constraints, or number of allowed routes per vehicle. These problem characterizations can be implemented alone or in combination with each other, thus making the number of different VRP problem instances significant. In this research, we limit the VRP problem characterization discussion to multiple depots, multiple nonhomogeneous vehicles, and route length restrictions.

In Laporte (1992), an overview of exact and approximate algorithms used to solve the vehicle routing problem is presented. The two specific instances of the VRP presented in the paper are the capacity constrained vehicle routing problem (CVRP) and distance constrained vehicle routing problem (DVRP). The exact algorithms for the VRP can be classified into three categories: direct tree approaches, dynamic programming, and integer linear programming. One direct tree approach is the *assignment lower bound and related branch and bound algorithm* by Laporte and others (1986). This method solved CVRPs with up to 260 vertices to optimality. Another direct tree approach is the *k-degree center tree and a related algorithm* by Christofides and others (1981). This method solved VRPs with up to 25 vertices to optimality. A dynamic programming approach called *state-space relaxation* by Christofides and others (1985b) solved CVRPs with up to 50 vertices to optimality. A *set partitioning and column generation method* to solve VRPs was first introduced by Balinski and Quandt (1964) and was later used to solve VRPs with time windows containing up to 100 vertices to optimality (Desrochers and others, 1991). A *two-index vehicle flow formulation* using a *constraint relaxation*

algorithm was introduced by Laporte and others (1985). This algorithm solved loosely constrained VRPs with up to 60 vertices to optimality.

The approximate approaches to solving the VRP presented by Laporte are the Clarke and Wright (1964) algorithm, the sweep algorithm, the Christofides-Mingozzi-Toth two-phase algorithm (1979), and a tabu search algorithm.

Laporte concludes his paper stating that the exact algorithms can only solve relatively small problems whereas the approximate algorithms can solve much larger problems with very satisfactory results. He believes that tabu search deserves greater attention as a means to solve large VRPs and achieve satisfactory solutions.

In a more recent article (Laporte and others, 2000), a survey of heuristic methods to solve the VRP is presented. The survey compares classical and modern heuristics and provides solution and computational time results. The classical heuristics presented include the *savings algorithms* originally developed by Clarke and Wright (1964), the *sweep algorithm* popularized by Gillett and Miller (1974), the *petal algorithms* first developed by Balinski and Quandt (1964), *cluster-first, route-second algorithms* where the best known algorithm is by Fisher and Jaikumar (1981), and a series of *improvement algorithms*. Computational results (using the 14 benchmark instances of Christofides *et al.* (1979)) show that the Clarke and Wright *savings* and Gillett and Miller *sweep algorithms* obtain good solutions quickly. The authors went on to state that the results of the classical heuristics did not come close to the results of the tabu search metaheuristic.

Laporte and others (2000) then present and compare different tabu search heuristics developed over the past few years. These include the *taburoute algorithm* of Gendreau *et al*. (1994), *Taillard's algorithm* by Taillard (1993), *Xu and Kelly's algorithm*

by Xu and Kelly (1996), *Rego and Roucairol's algorithm* by Rego and Roucairol (1996), *the adaptive memory procedure of Rochat and Taillard* developed by Rochat and Taillard (1995), and *the granular tabu search (GTS) algorithm* by Toth and Vigo (1998). Laporte and others do not rank order the algorithms by results. Instead, they provide the advantages and disadvantages of each method. They do state that the GTS algorithm by Toth and Vigo provided excellent solutions within very short computing times. They concluded that most tabu search algorithms can solve medium sized problems to near-optimality, and it is time to concentrate on methods that can solve larger instances. This would require faster, simpler and more robust algorithms.

Recent work not included in the survey includes Kelly and Xu's (1999) set partitioning based tabu search heuristic for the VRP. Their method uses a two-phased approach. The first phase generates unique routes using simple construction and improvement algorithms. The second phase combines the routes from the first phase and uses a set partitioning heuristic to find the best routes. The Xu and Kelly (1996) algorithm in the Laporte *et al.* (2000) survey only constrained the vehicle capacity. The formulation for the set-partitioning algorithm also constrains the route length.

In an article by Barbarosoglu and Ozgur (1999), a tabu search algorithm (DETABA) is developed to solve a single depot VRP. The algorithm uses concepts developed in previous tabu search algorithms, but modifies the intensification stage and neighbor construction procedures of the tabu search. The DETABA outperformed all but Taillard's (1993) VRP algorithm for test problems 1-5 and 11-12 of Christofides *et al.* (1979). This article did not compare DETABA to the GTS algorithm by Toth and Vigo (1998).

Another method to solve the VRP is presented by Baker and Sheasby (1999). Although their algorithm does not beat or match the best-known solutions for benchmark problems, they do provide satisfactory results much faster than previous algorithms. They use a method first developed by Fisher and Jaikumar (1981) that involves solving the VRP as a generalized assignment problem (GAP) and then as a traveling salesman problem. Baker and Sheasby's algorithm differs from Fisher and Jaikumar's algorithm in the manner they use the GAP solution. Instead of just using the optimal GAP solution like Fisher and Jaikumar, they use a number of near optimal GAP solutions before solving as a TSP. Solutions were within 1.59% of the best-known solutions. Although the solutions were not as satisfactory as best known solutions, they were computed in much less time.

2.2.2 Vehicle Scheduling Problems

Vehicle scheduling problems are routing problems in which additional constraints are added to consider the times when various activities may be carried out (Bodin and others, 1983). For example, if customer locations require delivery between certain time periods, then the problem now becomes a scheduling problem most commonly known as vehicle routing problems with time windows (VRPTW). The sequencing of vehicles in both space and time is the nature of the vehicle scheduling problem (Bodin and others, 1983).

Similar to the VRP, vehicle scheduling problems have many instances. The vehicle scheduling problem may also include single or multiple depots, homogeneous or nonhomogeneous vehicles, and any other instances of the vehicle routing problem with

the addition of time constraints. Bodin and others (1983) present a number of VSP instances including: single depot VSP, single depot VSP with length of path restrictions, single depot VSP with nonhomogeneous vehicles, and the multiple depot VSP.

There is extensive literature on the vehicle scheduling problem. The literature most pertinent to this research involves versions of the vehicle routing problem with time window constraints (VRPTW). Recent work on the VRPTW includes Carlton and Barnes (1996), Potvin and others (1996), Chiang and Russel (1997), and Liu and Shen (1999). Other variants of the VRPTW includes work by Desaulniers and others (1998) on the multiple depot vehicle scheduling problem with time windows (MDVSPTW), work by Hong and Park (1999) on the bi-objective VRPTW, work by Brandao and Mercer (1997) and Rodriguez and others (1998) on the multi-trip VRPTW, and work by Gendreau and others (1999) on a dynamic VRPTW. Some of the seminal work in the area of VRPTW includes Savelsbergh (1985), Solomon (1987), Solomon and others (1988), Solomon and Desrochers (1988), Desrochers and others (1992).

The temporal factor most relevant to this research is time window constraints. Time windows are defined as the earliest and latest time a customer allows the vehicle to provide service at its location. This is a very practical aspect for many VRP applications. Time windows can be hard or soft constraints. Hard time window constraints do not allow delivery time before the earliest arrival time or after the latest arrival time. Soft time window constraints allow the earliest and /or latest time periods to be violated, but there is usually a cost for these violations.

There are many instances of the VRPTW solved by different methods. The following is a presentation of a few of the instances related to the TDVRSP. These instances are solved using tabu search.

Potvin and others (1996) develop a tabu search heuristic for the vehicle routing problem with time windows. Their VRPTW has one central depot and a fleet of homogeneous vehicles. The time windows are hard, where arriving late is not allowed. Arriving early incurs a wait time penalty. The tabu search is based on specialized local search heuristics that maintain the feasibility of the solution at all times. The heuristic maintains feasibility because they state that it is difficult to get back to a feasible solution after an infeasible move (due to the time window constraints).

The tabu search heuristic uses both the 2-opt and Or-opt neighborhoods to find solutions to the VRPTW. The 2-opt exchange is from the k-opt family developed by Lin (1965). Here, *k* links are removed from a route and replaced by *k* new links in order to create a new route. Unfortunately, this move is not well suited for problems with time windows because they do not preserve the orientation of the routes. In this paper, the authors use the 2-opt exchange for multiple routes in a manner that preserves the orientation of the routes. Here, one link is removed from each route. Then the first customers on the first route are linked to the last customers on the second route, and vice versa.

The Or-opt exchanges are used to move a sequence of one, two, or three customers as intra or inter route exchanges. These work well as a move method for problems with time windows because it maintains the route order.

The neighborhoods for this algorithm use both the 2-opt and Or-opt moves. The 2-opt neighborhood considers every pair of links, where the links must be from different routes. For the Or-opt, every sequence of three customers, two customers, and one customer is considered and, for each sequence, all insertion places are also considered.

The tabu search prevents cycling by keeping a tabu list of the 2-opt and Or-opt exchanges. The inverse modification is maintained as tabu. The tabu move is overridden given appropriate aspiration criteria. Aspiration criteria are a user-defined criterion that permits the search to use tabu moves.

To save computation time, the move neighborhood is restricted to customers within a specified distance of each other. This confines the neighborhood to the most promising exchanges. This is expressed as follows:

a. Or-opt: customer i is moved between customers j and $j+1$ if customer i is one of the h nearest neighbors in distance from customer j.

b. 2-opt: links (i, $i+1$) and (j, $j+1$) are replaced by links (i, $j+1$) and (j, $i+1$) if customer $j+1$ is one of the h nearest in distance from customer i.

The number of nearest neighbors within a specified distance j is a parameter of the algorithm. The capacity and time windows must be satisfied in order for customer j to qualify as a neighbor in distance for customer i.

The special aspect of this algorithm is the altering between move neighborhoods. The algorithm begins using the 2-opt exchange. When a specified number of iterations are performed without any improvement to the solution, then the algorithm alternates to the Or-opt exchange. The process is repeated until the algorithm terminates. The construction phase of this algorithm uses Solomon’s I1 heuristic (Solomon, 1987).

In Chiang and Russell (1997), the authors use a reactive tabu search metaheuristic for the VRPTW. The reactive tabu search dynamically varies the size of the list of forbidden moves to avoid cycles and overly constrained search paths. They also use the ?-interchange mechanism of Osman (1993) as a neighborhood structure for the search process.

The VRPTW in the problem considered by Chiang and Russell has capacity and route constraints. The time windows are hard for late arrivals, and wait times are assigned to early arrivals. This problem has multiple objectives. The goal is to minimize the number of required vehicles, total travel time, and total travel distance in that prioritized order.

The reactive tabu search (RTS) methodology developed by Battiti and Techiolli (1994) is a means to strengthen the tabu search process. The idea is to dynamically vary the tabu list length during the search process. The purpose of reactive tabu search is to avoid local minima, which are fixed points that can trap a local search process, avoid limit cycles, which are closed orbits where the search process repeats as a sequence of solutions, and avoid chaotic attractors, which are a contraction of the search space so that the search process only visits a limited part of the solution space.

Chiang and Russell’s algorithm uses a parallel construction procedure, where customers are incrementally inserted to the best available route. The routes are initialized using the seed point generation scheme of Fisher and Jaikumar (1981), where seed points represent fictitious customers on a route that are deleted as soon as one real customer is added to the route. The customers are selected for route insertion based on three rules for ordering (to achieve time window feasibility). Rule one is based on the smallest early

time window parameter. Rule two is based on the tightness of the time window as calculated by 100(*l*-*e*)-*d*, where *l* is the latest time service can begin and *e* is the earliest time service can begin and *d* is the timed distance from the depot to the customer. Rule three is based on the largest value of *d*. The algorithm performs six passes evaluating where the customer should be inserted based on the three rules and two forms of evaluation (distance added and time added to the route).

During the construction phase, improvement procedures are embedded in the construction process. It is invoked each time another 10 percent of the customers have been added to the emerging routes. It is invoked one final time when all the customers have been added to routes.

The improvement process proceeds iteratively from one solution to another until a termination step is induced. Improvement is conducted by using the ? -interchange mechanism. This move method is an ordered search method that examines all possible combinations of pairs of routes for exchange. The method either moves a customer from one route to another or exchanges customers between routes.

The algorithm also has methods to evaluate time window constraints for move mechanisms. The methods are the push forward and push back calculations. These efficiently determine the customer service start times as a consequence of a move on a solution.

For the purposes of intensification, the authors use an elite solution list. For diversification, the authors use frequency-based memory in order to prevent solutions from frequent switching of customers.

Reactive tabu search maintains a history of each solution. The history includes solution attributes such as number of vehicles, total travel time, and others. If a new feasible solution is found, it is checked against the solutions in the RTS memory. If the solution does not already exist, it is added to the RTS memory. If the solution already exists, a repetition variable is increased documenting the repetition. If the repetition occurs more than a specified number of times, the solution is moved to the often-repeated set. If the often-repeated set of solutions is more than a specified tolerance, the tabu list size is increased. If no feasible solutions are found in the tabu search, the tabu list size is decreased.

In Brandao and Mercer (1997), a tabu search approach is developed for a real world VRPTW with similarities to the TDVRSP. In this problem, they solve a VRPTW with the following characteristics: customer time windows, nonhomogeneous vehicles with different capacities, ability to hire additional vehicles as necessary, restricted access to some customers, drivers' scheduling constraints, unloading times, and multiple trips per vehicle. The multiple trips per vehicle characteristic is a very important additional feature that is very common in practice but scarcely studied.

The authors mentioned that Taillard (1996) studied the multiple trips per vehicle in a VRP. In that algorithm, Taillard generated a large set of vehicle routes that satisfy the VRP constraints. It then selects a subset of routes by enumeration, which is assembled into feasible working days using several applications of a bin-packing algorithm. Brandao and Mercer (1997) use a much different algorithm in order to account for the time windows and other real world constraints.

The tabu search multiple trip vehicle routing and scheduling problem (TSMTVRSP) algorithm consists of three phases applied sequentially, where the initial solution of each phase is the best solution of the previous phase.

In the first phase, routes are constructed based on the nearest neighbor principle and method of insertion, where the customer from the set of nearest neighbors is inserted into a route and evaluated. This phase uses insert moves to create routes and swap moves between the routes as trial moves.

Phase two attempts to make solutions feasible in terms of routing and scheduling constraints, and to reduce the solution costs. The trial moves in phase two are the insert and swap moves. The insert move removes a customer from one route and inserts it into another route. The insert move is based on the GENI algorithm developed by Gendreau (1992). If the route is a new route, then a vehicle is assigned to the route; if there are no existing vehicles, then a new vehicle is hired. Creating new routes help make the solution feasible. Swap moves exchange customers between routes. Each trial solution is evaluated by two equations that consider scheduling constraints and infeasibilities.

Phase three is very similar to phase two. The difference is that in phase three, the solutions are not allowed to go infeasible when searching for new solutions. The evaluation in phase three is based on cost, whereas the evaluation in phase two was based primarily on time.

In Cordeau and others (1997), the authors develop a simple and robust tabu search heuristic capable of solving three instances of the VRP. The instances are the periodic vehicle routing problem (PVRP), periodic traveling salesman problem (PTSP), and multi-depot vehicle routing problem (MDVRP).

The PVRP is characterized by a planning horizon and multiple customer visits within the horizon. The problem consists of simultaneously selecting a visit combination for each customer and establishing vehicle routes for each day of the planning horizon, according to the VRP constraints. Vehicles may only make one trip per day. The PTSP is a special case of the PVRP, where only one vehicle is considered and no demands are specified per customer. The MDVRP is defined on a single day where vehicles operate out of multiple depots sites. Each vehicle starts and ends its route at the same depot. This algorithm does not consider scheduling constraints and the objective of each problem is to minimize the total travel costs.

The formulation of the PVRP is presented in Cordeau and others (1997) and is similar to Bodin and others (1983). In this case, one additional index was added to the decision variable x^v_{ij}. The additional index l represents the day of travel for vehicle v. An additional constraint was added that guarantees each customer is visited only on the days corresponding to the specified visit days. The authors formulate the MDVRP simply by defining index l to specify the depot instead of the period.

The construction phase for the initial solution of the algorithm uses GENI to create the routes. The GENI heuristic is a means to insert un-routed customers or to remove customers from their current routes and reinsert them into different routes. The GENI heuristic constructs Hamiltonian tours by inserting at each step a vertex v between two of its p closest neighbors v_i and v_j.

The improvement phase consists of searching for new solutions and evaluating the cost with incurred penalties of each trial solution. The new solutions are determined by evaluating a series of insert moves on the current solution. The neighborhood is

composed of all solutions that can be obtained by performing one of the following transformations:

1. Remove customer i from route k on day l and insert it into another route k'
2. (a) Replace visit combination r currently assigned to customer i with another combination $r' \in Ci$
   (b) For $l = 1, \ldots, t$, do
      i. If $a_{rl} = 1$ and $a_{r'l} = 0$, remove customer i from its route on day l
      ii. If $a_{rl} = 0$ and $a_{r'l} = 1$, insert customer i into the route on day l minimizing the increase in f(s), where f(s) is the objective function.

The algorithm developed in this paper was applied to a number of instances in the literature for three problem types: PVRP, PTSP, and MDVRP. They concluded their algorithm outperformed previous results in the literature for the three problem types.

In Gendreau and others (1999), a heterogeneous vehicle routing problem is solved using a tabu search heuristic. Heterogeneous vehicles in this problem have various capacities with fixed and variable costs. Their tabu search heuristic produced high quality solutions, including several new best solutions, on a set of benchmark problems. The objective of the algorithm is to find the best fleet composition given an unlimited number of vehicles rather than making the best possible use of a given fleet.

In their algorithm, the authors use the GENIUS heuristic by Gendreau (1992) in the construction of an initial solution. The GENIUS algorithm consists of a tour construction phase called *generalized insertion* (GENI) and of an improvement phase called *unstringing and stringing* (US). Routes are incrementally constructed by GENI and improved by US between GENI steps. The neighborhood structure is defined by a random selection of a subset of customers for insertion.

The improvement phase uses an exchange procedure that swaps two vertices belonging to two neighbor routes and a fleet change procedure, which swaps sub routes between routes. The exchange procedure is an intensification effort whereas the fleet change procedure is a diversification device.

The authors also use an adaptive memory procedure; also known as probabilistic diversification and intensification. That procedure works with a pool of partial and full solutions in a constantly updated memory and is used as an initial solution generator. At each step, the procedure combines the best features from each partial solution to generate new solutions, which are then improved upon through the local search process.

In Kim and Kim (1999), the authors solve a multi-period vehicle scheduling problem (MPVSP), where a fleet of homogeneous vehicles delivers a single product from a central depot to multiple customers over multiple time periods. This study has some similarities to the TDVRSP in which vehicles make multiple trips and customers receive multiple services over the course of the cumulative time periods. The difference between the two is that the MPVSP is decomposed into multiple discrete time periods. Within each time period a vehicle may only make one trip and a customer may only be serviced one time. The customers' total demand is no more than a single vehicle's capacity. The time needed for a one-way trip between the depot and a retailer is a multiple of a time period and is the same regardless of the direction of the trip (Kim and Kim, 1999). The objective is to minimize the vehicular transportation costs and customer inventory holding costs.

The algorithm used to solve the MPVSP is performed in two phases. The first phase decomposes the MPVSP into N single retailer problems by ignoring the number of

vehicles on hand. Each single retailer problem is considered a multiperiod lot sizing problem with time –varying demands, in which the exact requirement policy is used. The problem is represented as a shortest path problem, where a path denotes a delivery schedule. The algorithm generates several possible shortest paths for each customer. In the second phase, a delivery schedule is selected for each customer among the candidate delivery schedules generated in the first phase. A heuristic is used to find a set of feasible delivery schedules (Kim and Kim, 1999).

2.3 Summary

The literature for the General Vehicle Routing Problem (GVRP) and its extensions is significant. Forms of the GVRP have been solved with a myriad of methods over the years. In many recent studies, the tabu search technique was used quite successfully as a means to find good quick solutions. The literature reveals that tabu search techniques are finding better solutions quicker than conventional optimization techniques for large GVRP problems.

Most all studies have concentrated on traditional GVRP problems that include route length constraints and time windows. There are few publications dealing with the multiple trips dimension or multiple services dimension. There are no publications that consider the multiple trips and multiple services dimensions in a single time period. There are also no publications that consider the multiple trips, multiple services with hub dimensions.

III. Group Theoretic Application To Tabu Search

3.1 Introduction

Colletti (1999) presents group theory as a unifying mathematical framework for the study of metaheuristics. In his dissertation, he presents a number of group theoretic concepts for building metaheuristic move methods, move neighborhoods, and other strategies of combinatorial optimization (Colletti, 1999). The specific group used in Colletti's research and this research is the symmetric group on n-letters, denoted S_n.

Chapter 3 presents the basic concepts of group theory and its application to the TDVRSP and tabu search. In an effort to satisfy both the Operations Research and Algebraist communities, the concepts are presented using language and notation from each. A goal is to bring the communities together in understanding how group theory and metaheuristics complement each other.

Section 3.2 introduces groups and S_n. Section 3.3 explains how the TDVRSP is formulated as S_n. Section 3.4 describes how S_n partitions the TDVRSP solution space using conjugacy classes and orbits. Section 3.5 introduces the concept of orbital planes, which are used as a means to further partition the TDVRSP solution space. Section 3.6 presents the solution space partition hierarchy. Section 3.7 describes how the solution space is traversed using the group theoretic tabu search (GTTS) approach.

3.2 Groups and The Symmetric Group on n-Letters

Section 3.2 provides basic definitions and examples for groups and S_n. This information is used in later sections to describe other group theoretic concepts essential to GTTS.

3.2.1 Groups

Abstract groups are simply explained as sets of objects, together with a method of combining its elements that is subject to a few simple rules (Baumslag and Chandler, 1968). Within the family of abstract groups is the semi-group and group.

Definition (*semi-group*) A non empty set G together with a fixed binary operation $\oplus$ that satisfies the following conditions:

1. $x \oplus y \in G, \forall x,y \in G$; the operation is closed
2. $(x \oplus y) \oplus z = x \oplus (y \oplus z) \ \forall x,y,z \in G$; the operation is associative

Definition (*group*) A semi-group that satisfies the additional conditions:

1. $\exists ! e \in G \ni \forall x \in G, e \oplus x = x \oplus e = x$; there exists a unique identity
2. For each $x \in G, \exists ! x^{-1} \in G \ni x^{-1} \oplus x = x \oplus x^{-1} = e$; there exists a unique inverse

An example of a group is the set X={2,4,6,8}, where $\oplus$ is the mod($x*y$/10) and x, $y \in$ X (Barnard and Neill, 1996). The results of $x \oplus y$ are displayed in Table 3.1. Note $\oplus$ is associative and closed within X. There also exists one row exactly reflecting the top row and one column exactly reflecting the left hand column, which is a result of the

identity property where 6 is the identity element. The number 6 is in each column and row, which shows each element has an inverse.

Table 3.1 Example of a Group

$\oplus$	**2**	**4**	**6**	**8**
2	4	8	2	6
4	8	6	4	2
6	2	4	6	8
8	6	2	8	4

Groups have some elementary properties that are used in this research. These properties, presented as a theorem with a proof in Barnard and Neill (1996) are presented below.

Properties of group G.

1. For $x, y \in G$, if $x \oplus y = e$, then $x = y^{-1}$ and $y = x^{-1}$
2. $(x \oplus y)^{-1} = y^{-1} \oplus x^{-1}$ $\forall x, y \in G$
3. $(x^{-1})^{-1} = x$ $\forall x \in G$
4. For $x, y, z \in$ G, if $z \oplus x = z \oplus y$, then $x = y$ and if $x \oplus z = y \oplus z$ then $x = y$

A group that is used throughout this research is the symmetric group on n-letters, S_n. The properties presented for groups apply to S_n. S_n is further described in Section 3.2.2.

3.2.2 Symmetric Group on n-Letters

Since the TDVRSP is essentially a combinatorial optimization problem involving the permutation of letters, the symmetric group on n-letters provides a natural setting for structuring the problem.

Definition (*permutation of a set A*) is a function from A into A, which is both one to one, and onto (Fraleigh, 1976).

Definition (*symmetric group on n-letters, S_n*) is the group of all permutations of set A if A is the finite set {1, 2, 3, … , n}(Fraleigh, 1976).

There are two different notations for S_n, the standard form and cyclic form. The **standard form notation** is a 2 by n array that represents a one to one and onto function whose domain (top row) and image (bottom row) are the integers $\{1,2,\ldots,n\}$ (Colletti, 1999). Let elements $\pi \in S_n$ be permutations. The notation is the array

$$\begin{matrix} 1 & 2 & 3 & 4 & 5 & \ldots & n \\ \pi(1) & \pi(2) & \pi(3) & \pi(4) & \pi(5) & \ldots & \pi(n) \end{matrix} .$$

The **cyclic form notation** is a streamlined notation of S_n. Given $i \in \{1,2,\ldots,n\}$, $p \in \mathbf{N}$ and $\pi^p(i) = i$, the first cycle is represented as:

$$(i, \pi(i), \pi^2(i),\ldots,\pi^{p-1}(i)).$$

Equivalently, the **cycle** (i, j, k, l) means π sends i to j, j to k, k to l, and l back to i. The process continues by picking an element not in the cycle containing i and iterating the process until all members of $\{1,2,\ldots,n\}$ have been used. A cycle of length k is a cycle containing k elements (Sagan, 1991). Cycles with only one element are called **unit** cycles, where unit cycles can be implied and dropped from cyclic notation. An

involution is a permutation such that $\pi^2 = e$. Consequently, all cycles with lengths 1 and 2 are involutions.

Examples of the standard form permutation and cyclic form permutation are presented below (Sagan, 1991).

If $\pi \in S_5$ is

$\pi(1) = 2$, $\pi(2) = 3$, $\pi(3) = 1$, $\pi(4) = 4$, $\pi(5) = 5$,

then the standard form is $\pi = \begin{pmatrix} x \\ p(x) \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 3 & 1 & 4 & 5 \end{pmatrix}$

and the cyclic form is $\pi = (1,2,3)(4)(5)$ or $\pi = (1,2,3)$.

The length of the first cycle is 3 and the unit cycles are (4) and (5).

The symmetric group on n-letters binary operation is function composition. The product of two permutations π and σ, denoted $\pi \oplus \sigma$, is composition. That is, $\pi \oplus \sigma = \pi \circ \sigma$. Given $x \in S_n$, then

$$(\pi \oplus \sigma)(x) = \sigma(\pi(x)).$$

For example, let $\pi = \begin{pmatrix} x \\ p(x) \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 3 & 1 & 4 & 5 \end{pmatrix}$ and $\sigma = \begin{pmatrix} x \\ s(x) \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 4 & 3 & 1 & 2 & 5 \end{pmatrix}$

then

$$\pi \oplus \sigma = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 2 & 3 & 1 & 4 & 5 \end{pmatrix}\begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 4 & 3 & 1 & 2 & 5 \end{pmatrix} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 \\ 3 & 1 & 4 & 2 & 5 \end{pmatrix}.$$

An operation used throughout this research as a means to execute tabu search moves is conjugation. Conjugation provides a relabeling of permutation letters while maintaining the original cyclic structure.

Definition (*conjugation*) an operation, denoted $x = y^x$, where $x = k^{-1} \oplus y \oplus k$ for some x, y, $k \in G$ where G is a group.

Although conjugation looks like the power operation, it is not because the exponent is a group element and not an integer (Colletti, 1999).

An example of the conjugation operation as it is applied later, is the operation between two permutations where one represents an incumbent solution and the other a two-letter swap move. The incumbent solution and two-letter swap move are represented as x = (1,3,2)(4)(5) and k = (2,4), respectively. Conjugating x by k results in the solution $x^k = k^{-1} \oplus x \oplus k$ = (1,3,4)(2)(5), where the cycle structure is maintained and only letters in the conjugator permutation are moved.

The definitions and concepts presented for groups and the symmetric group on n-letters is a simple introduction to group theory. Other concepts such as conjugacy classes, transpositions, orbits, cosets, double cosets, transversals, and centralizers are presented later as needed.

3.3 TDVRSP in Terms of S_n

Section 3.3 begins by mapping a general VRP formulation to the symmetric group VRP formulation in Section 3.3.1. Next, the TDVRSP is formatted as a symmetric group object in Section 3.3.2.

3.3.1 Formulating a VRP in S_n

Formulating vehicle routing problems consists of an objective function, *f*, and constraints. Generally, the objective is to minimize the distance and/or time of a route, *r*, from a set of feasible routes, *R*. The route is a collection of edges, a_{ij}, that connect vertice *i* to vertice *j*.

$$\begin{gathered}\text{Minimize } f(r)\\ r \in R\\ r = \{a_{ij} : a_{ij} = 1\}\\ a_{ij} = 0,1\end{gathered}$$

Feasible routes are cyclic digraphs, Figure 3.1, where each ordered pair of vertices occurs at most once as an edge. The route is a closed path starting and ending at the same vertex

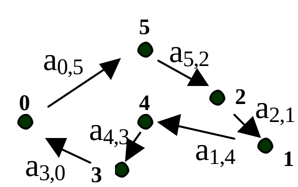

Figure 3.1 Cyclic Digraph Example

The cyclic digraph is mapped to an adjacency matrix where each edge of the graph is an entry. Each column and row has a unique entry. Adjacency matrices with exactly one 1 in each row and column is a permutation matrix. Therefore, the

formulation converts to finding the best adjacency matrix, **A**, among the set of permutation matrices, **P**.

$$\underset{\mathbf{A}\in \mathbf{P}}{\text{Minimize}}\ f(\mathbf{A})$$

Table 3.2 Adjacency Matrix for Figure 3.1

	0	**1**	**2**	**3**	**4**	**5**
0						1
1					1	
2		1				
3	1					
4				1		
5			1			

Permutation matrices are mapped to S_n, where the domain is the row headings and the range is the column headings for each entry. Permutation x is as an example of the matrix shown in Table 3.2.

$$x = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 \\ 5 & 4 & 1 & 0 & 3 & 2 \end{pmatrix}$$

The formulation of a VRP using symmetric groups is noted below.

$$\underset{x\in\ \Omega \subseteq S_n}{\text{Minimize}}\ \tilde{f}\,(x)$$

The symmetric group formulation has an objective function, $\tilde{f}$, that maps a permutation, x, to an objective function value. Permutations are constrained by Ω, which is a subset of S_n. The subset Ω is a collection of $x\in S_n$, where S_n is constrained by conjugacy classes and other restrictions. The group theory presented in this dissertation determines Ω, and discusses how it is partitioned and searched.

3.3.2 Formatting a TDVRSP in S_n

In order to apply group theoretic tabu search to the TDVRSP, solutions must first be formatted as a permutation. Using information provided on S_n in Section 3.2.2, TDVRSP solutions are configured as permutations in disjoint cyclic form. Although the TDVRSP can be configured in standard permutation form, the disjoint cyclic form provides better visual comprehension of the solution and more efficient computing measures.

The disjoint cyclic form is composed of multiple-letter cycles and single-letter cycles called unit cycles, e.g. permutation (1,2,3)(4)(5) in S_5. By convention, we omit unit cycles and call those that remain the permutation's "factors." Each disjoint cyclic form has a specific cycle structure dependent on the size and number of factors. For example, a permutation with two 3-cycles has the cycle form structure (x,x,x)(x,x,x), also noted as 3^2.

For a TDVRSP solution, each factor represents a trip. In this example, the 3-cycle represents the following trip: $1 \rightarrow 2 \rightarrow 3 \rightarrow 1$. For the TDVRSP, the first letter in the cycle represents a vehicle trip letter. Letters in positions 2, 3, … , *m*, represent the customer letters serviced by that vehicle trip letter. In this example, vehicle trip letter 1 leaves a depot/hub and services customer letters 2 and 3 before returning to the depot/hub. A unit cycle represents either a vehicle letter not leaving the depot/hub to conduct a trip or a customer letter not serviced within a trip. Permutations having a single factor represent TDVRSP solutions having one trip. Permutations with *k* factors represent *k* vehicle letters conducting *k* trips. The *k* vehicles could be from the same or different depots/hubs depending on the vehicle location data.

In group theoretic notation, let C and V be the disjoint customer and vehicle letter-sets, respectively, where $|V| < |C|$, and let X be the permutations in $S(C \cup V)$ whose factors each contain a sole V-letter. The *first position* of any cycle in X is that of its single V-letter, and the factors of $x \in X$ are arranged in ascending V-letter order, thus implying lexicographic ordering.

Because Wiley's (2001) Java™ class file for S_n structures cycles by lexicographic ordering, all vehicle letter labels precede customer letter labels. This also provides more structured letter accountability. For example, if there are 4 vehicle trip letters and 4 customer service letters in S_8, then letters 1,2,3,4 represent the vehicle trip letters and letters 5,6,7,8 represent the customer service letters.

In a TDVRSP problem, there are vehicles that make multiple trips within a time period and customers that receive multiple services within a time period. A letter is allocated for each vehicle trip and each customer service. Letters are assigned sequentially for each vehicle or customer. For example, if vehicle *A* has the potential to make three trips, letters 1,2,3 may be assigned to vehicle *A*'s trips. If customer *A* has the potential to receive three services, then letters 4,5,6 may be assigned to each customer service. Table 3.3 shows a letter assignment to vehicles *A*, *B*, and *C* and customers *A*, *B*, and *C*. Vehicle *A* has the potential for 3 trips, vehicle *B* can make 2 trips and vehicle *C* can make 4 trips. Customer *A* has the potential for 2 services, customer *B* can be serviced 3 times, and customer *C* is only serviced up to 1 time.

Table 3.3 S_n Letter Assignment Example

	Number of trips/services	Assigned letters
Vehicle *A*	3	1,2,3
Vehicle *B*	2	4,5
Vehicle *C*	4	6,7,8,9
Customer *A*	2	10,11
Customer *B*	3	12,13,14
Customer *C*	1	15

The number of service letters assigned per customer depends on their demand requirements and vehicle capacities. The model user is required to specify the number of customer service letters for each customer. The number of letters is a balance between providing enough letters as service placeholders and providing too many, which consequently increases problem size.

A graphical example of a TDVRSP solution with the labeling scheme displayed in Table 3.3 is presented in Figure 3.2. For the solution, vehicle trips occur in numerical order but customer services do not necessarily occur in numerical order. The solution tour is (1,10)(2,11,13)(4,12)(6,15) where unit cycles (3), (5), (7), (8), (9) are unused vehicle trips and unit cycle (14) is an unused customer service letter.

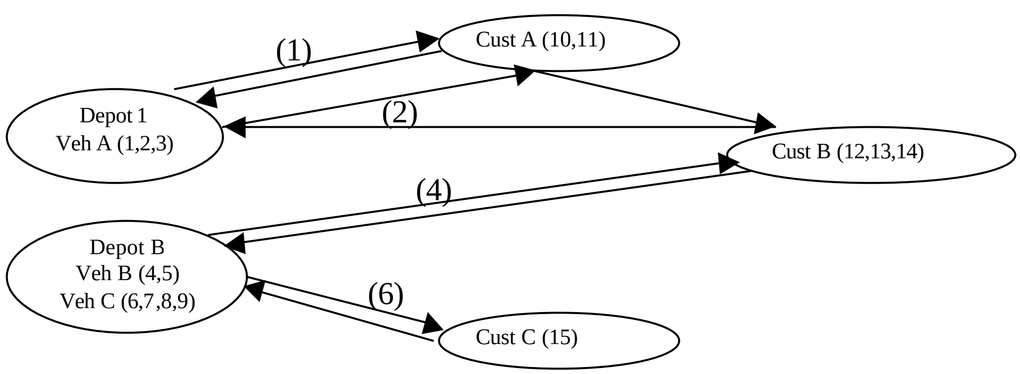

Figure 3.2 TDVRSP Solution Tour Example

Another example of a TDVRSP solution and its labeling scheme is presented in Figure 3.3. This TDVRSP instance includes hubs as a dimension. Including hubs requires a hierarchical letter assignment scheme for vehicle trips and customer services. The hierarchy is based on precedence relations within the distribution structure. Letters are assigned first to vehicles and customers that belong to higher precedence nodes. Vehicles and customers in the lowest precedence nodes are assigned letters last. All vehicle trip letters are assigned their letters before customer service letters are assigned.

In the TDVRSP, the highest precedence nodes are the depots and the customers they serve. The second highest precedence nodes are the customers served by the hubs in the highest order. The third highest precedence nodes are the customers serviced by the hubs in the second order, and so on. The term **tier** is used to denote the collection of customers served by a particular depot or hub. Note that customers *B* and *E* are hubs and have vehicles assigned.

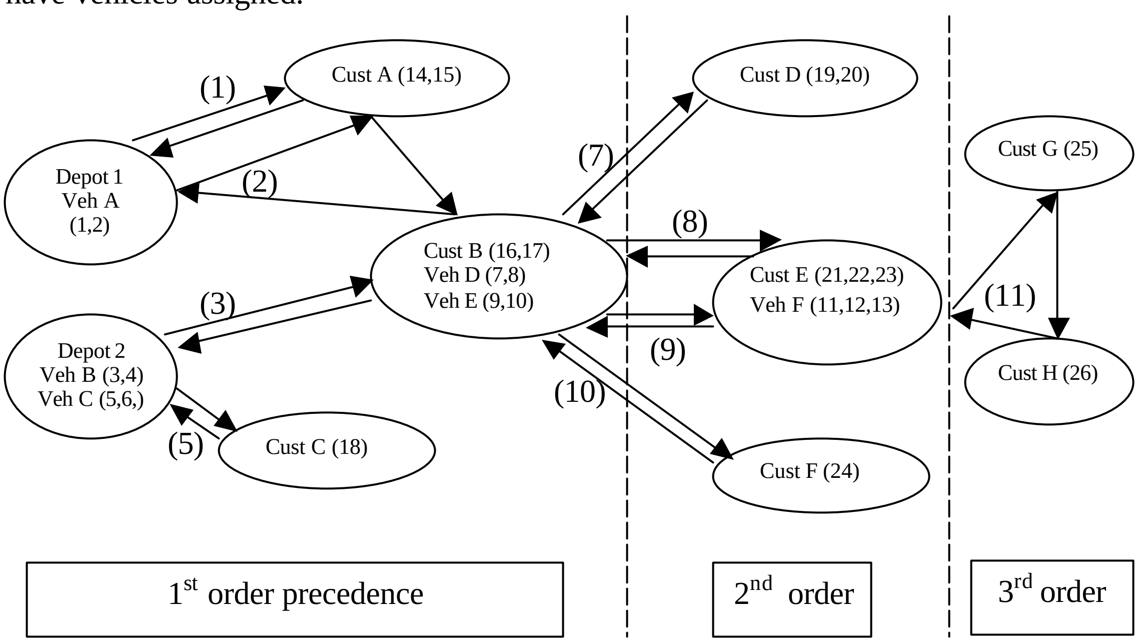

Figure 3.3 Solution Tour With Hubs Example

The disjoint cyclic tour represented by Figure 3.2 is

$$(1,14)(2,15,16)(3,17)(5,18)(7,19)(8,21)(9,22)(10,24)(11,25,26)$$

with unit cycles (4), (6), (12), (13) as unused vehicle trip letters and unit cycles (20) and (23) as unused customer service letters. Note that depots 1 and 2 and customers *A*, *B*, and *C* are the first order precedence. Their vehicles are assigned letters first. Customers *D*, *E*, and *F* are in the second precedence level and the vehicles assigned to customer *E* are allocated letters that follow those previously assigned. Customers *G* and *H* are in the last precedence level.

Once all vehicle trip letters are assigned, the customer service letters are allocated based on precedence level. The letters are allocated in the same manner as the vehicle letters. Higher precedence customers receive their customer service letters before the lower levels receive their letters.

3.4 Partitioning the Solution Space with S_n

Group theory has a major role in the proposed TDVRSP group theoretic tabu search algorithm. Group theory provides structure for the TDVRSP instance, tabu search moves, and move neighborhoods. It also provides an efficient means to search the solution space. This section introduces some group theoretic concepts that partition the solution space and provide a means to prevent cycling, which makes the search more efficient.

The number of possible solutions, the cardinality of the solution space, for S_n equals *n*!, which becomes large as *n* increases in value. For example, the number of possible solutions for S_8 is 40,320 and for S_{25} is 25!. The incredible size of S_n's solution

space makes its reduction for the tabu search essential. Fortunately, group theory provides a means to partition the solution space, thus enabling the concentration of search efforts in more effective regions of the space. This section describes two group theoretic features that partition the solution space.

Group theory provides for the partitioning of groups into conjugacy classes, group actions and orbits. In general, any abstract group G inherently self-partitions into conjugacy classes.

Definition (*conjugacy class*) the set of all elements $\{h^{-1} \oplus g \text{ Å } h : h \in G\}$ for $g \in G$. The conjugacy class of $g \in G$ is denoted as

$$\text{CClass}(G, g) = \{g^h : h \in G\}.$$

Additionally, conjugacy classes of any group are mutually exclusive and exhaustive (Colletti, 1999).

When $G = S_n$:

- Two permutations are in the same conjugacy class iff they share a cycle structure;
- The size of the conjugacy class having n_i i-cycles, where g is a specific cycle structure is

$$|\text{CClass}(G, g)| = \frac{n!}{\prod n_i! i^{n_i}}.$$

Note, if G is a proper subgroup of S_n, permutations $x,y \in G$ may not be conjugate even if they share a similar cycle structure (Colletti, 2001).

For abstract group G, the G-conjugacy class of $g \in G$ is also denoted g^G. Thus, a conjugacy class is *relative* to G. For instance, the conjugacy classes of S_n split along

cycle structure, while those of an alternating group on n-letters, denoted A_n, do not. A_n is a subgroup of S_n comprised of all even permutations of S_n (Fraleigh, 1976).

Figure 3.4 provides a visual example of S_8 and its 22 conjugacy classes. The conjugacy classes with their factors are listed below. Unit cycles, typically not noted, are included for better visualization of the conjugacy class cycle structure composition. CC*k* is a labeling system for CClass(S_8,x) $\forall\, x \in S_8$.

CC1 = 1^8	CC2 = $2^1 1^6$	CC3 = $2^2 1^4$	CC4 = $2^3 1^2$
CC5 = 2^4	CC6 = $3^1 1^5$	CC7 = $3^1 2^1 1^3$	CC8 = $3^1 2^2 1^1$
CC9 = $3^2 1^2$	CC10 = $3^2 2^1$	CC11 = $4^1 1^4$	CC12 = $4^1 2^1 1^2$
CC13 = $4^1 2^2$	CC14 = $4^1 3^1 1^1$	CC15 = 4^2	CC16 = $5^1 1^3$
CC17 = $5^1 2^1 1^1$	CC18 = $5^1 3^1$	CC19 = $6^1 1^2$	CC20 = $6^1 2^1$
CC21 = $7^1 1^1$	CC22 = 8^1		

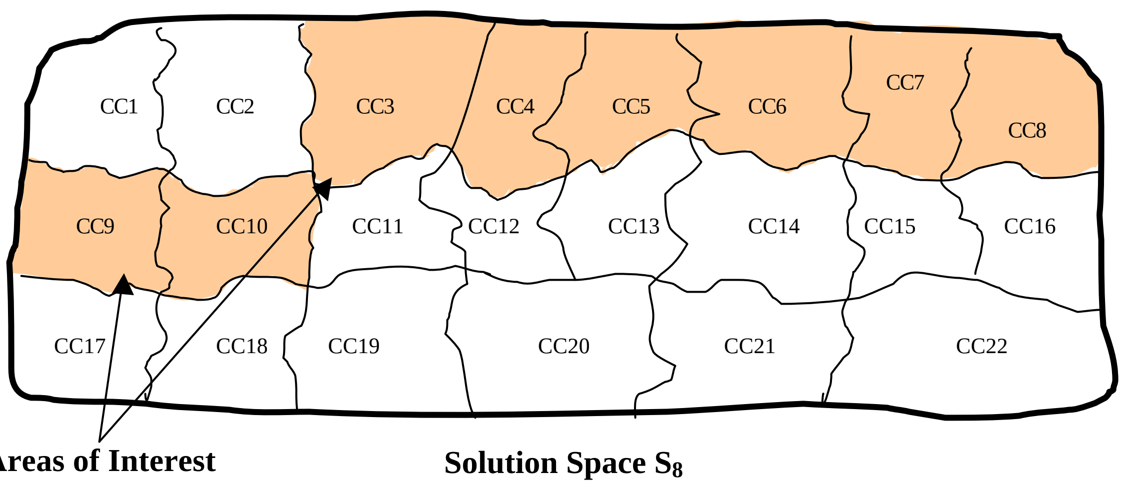

Figure 3.4 Example Solution Space With Conjugacy Class Partitions

The TDVRSP characteristics direct special cyclic structure ensuring the problem maintains feasibility. Therefore, some conjugacy classes in the solution space are not explored during the tabu search. For example, if a TDVRSP problem instance assumes vehicles serve at most two customers, the only feasible solutions are conjugacy classes

whose cycle structures consist of 2 and 3-cycles. Another guiding factor for the conjugacy classes is the number of customers that require service. For example, assuming there exist at least two customers requiring service, any conjugacy class with less than two 2-cycles or one 3-cycle is not considered. The number of conjugacy classes in S_8 is reduced to eight and are highlighted in Figure 3.4.

Reducing the number of conjugacy classes in the solution space reduces the number of solutions the search will traverse. The number of solutions in each screened conjugacy class is listed below. The total number of solutions is now 5,887 versus 40,320. The cardinality for each conjugacy class is

$$|CC3| = 210 \quad |CC4| = 420 \quad |CC5| = 105 \quad |CC6| = 112$$
$$|CC7| = 1120 \quad |CC8| = 1680 \quad |CC9| = 1120 \quad |CC10| = 1120$$

$$\text{Sum} = 5887$$

The partitioning of S_n by conjugacy classes provides a means to search only relevant portions of the solution space. This relieves the search from entering regions where no feasible solution exists.

Another feature of group theory that partitions S_n solution space are group actions and orbits. Given an abstract group G and a set X, a *group action*, denoted $_GX$ defines how G-elements operate upon X-elements to create X-elements (Colletti, 1999).

Definition (*group action*) Let G be a group and X be a non empty set. For $g \in G$ and $x \in X$, let x^g denote the unique X-element that satisfies:

- $x^e = x$, $\forall\, x \in X$ (e is the G-identity)
- $\forall\, g,h \in G$ and $x \in X$, $(x^g)^h = x^{gh}$

The "conjugation-like" notation $x^g \in X$ denotes the result of $g \in G$ acting upon $x \in X$. The user defines G and the result of x^g must be closed on X.

A group action inherently partitions X into *orbits.*

Definition (*orbit)* Let G be a group and X be a non empty set. For $x \in X$, an orbit is the set of all elements in X to which x can be moved by $g \in G$. i.e., $x^G \equiv \{x^g : g \in G\}$.

The *orbit* of $x \in X$ is x^G and denotes the G-orbit of x, Orbit (G, x). Because any two orbits of ${}_GX$ either coincide or are disjoint, we have that orbits of a group action partition X. If X is a conjugacy class in S_n, orbits will exhaustively and exclusively partition the conjugacy classes (Colletti, 1999).

Specifically for this research, the set $X \in$ CClass (S_n, x) and group G is a user-defined subgroup of S_n that moves customer letters. The G-orbit of $x \in X$ defines a tabu search move neighborhood of x. A particular x^g defines a move neighbor of x. Each orbit is the neighborhood of any one of its permutations, and the members of an orbit share the same neighborhood (Colletti, 1999). Neighbors and neighborhoods are further discussed in Chapter 4.

The number of orbits in ${}_GX$ is defined by the Cauchy-Frobenius Theorem (Isaacs, 1994):

$$|G|^{-1} \sum_{g \in G} fix(g)$$

where *fix(g)* is the number of X-elements that g fixes, i.e. the number of $x \in X$ such that $x^g = x$. Since $|G|$ and the number of orbits are inversely proportional, the former affects

anti-cycling methods used during the search. The size of G becomes a factor when determining how to best traverse the solution space during the tabu search.

The following example builds an orbit of ${}_GX$, where X is CC5 from S_8 and G is the sub-group of S_8 defined as:

$$\{(),\ (5,6)(7,8),\ (5,7)(6,8),\ (5,8)(6,7)\}$$

The orbit of $x = (1,5)(2,6)(3,7)(4,8) \in \text{CC5}$ is:

$x^G = \{(1,5)(2,6)(3,7)(4,8),\ (1,6)(2,5)(3,8)(4,7),\ (1,7)(2,8)(3,5)(4,6),\ (1,8)(2,7)(3,6)(4,5)\}$

This is also the orbit for each element in x^G.

3.4.1 Traversing Conjugacy Classes With Orbits

Given the information on S_n, conjugacy classes, and orbits presented thus far, there exists a method that exhaustively traverses the solution space of a conjugacy class. This method partitions the conjugacy class via orbits and generates a transversal list for systematically moving through the space.

Although useful for small problem sizes, this method becomes computationally expensive as the problem size increases. Methods that address larger TDVRSP problem sizes are presented in Section 3.5. However, this section is introduced first as a conceptual building block.

Provided a specific $x \in X$, the following describes how to traverse the orbits of a group action ${}_GX$, where G is a sub-group ($\leq$) of S_n, and X is a conjugacy class in S_n. Colletti (1999) provides a method that uses transversals of Double Cosets in order to create all the orbits. Each orbit is then explored.

In order to traverse orbits, the concepts of cosets, double cosets, centralizers, and transversals are introduced.

Definition (*cosets*) For $B \leq G$ and $g \in G$, the sets: $gB \equiv \{gb : b \in B\}$ and $Bg \equiv \{bg : b \in B\}$ are left and right cosets of B in G, respectively.

The collection of all left (right) cosets partition G. For finite G, the number of left (right) cosets equals $|G| / |B|$.

Definition (*double cosets*) Given $J \leq G$, $K \leq G$ and $g \in G$, the set $JgK = \{jgk: j \in J, k \in K\}$ is a double coset of "J and K in G."

The collection of all such double cosets:

$$\text{DCosets}\ (G, J, K) \equiv \{JgK : g \in G\}$$

partitions G (Dixon,1973). Colletti (1999) shows that the number of orbits of a group action is tied to the concept of double cosets.

Definition (*centralizers*) For $H \leq G$ and $g \in G$, the centralizer of g in H – denoted Cent (H, g) – are all the H-elements that commute with g, i.e. $gh = hg$ or $h^{-1}gh = g^h$.

Centralizers are special types of point stabilizers (Colletti, 1999) and can be used to find orbital transversals.

Definition (*transversals*) a transversal is a collection of elements from disjoint sets.

Simply put, if $\{X_i\}$ is a collection of disjoint sets, then a transversal is a set of elements, one from each X_i (Colletti, 2001).

Colletti (1999) provides an example on how to build orbital transversals T. Given the group action ${}_HX$, where $H \leq S_n$ and $x \in X$ is a conjugacy class in S_n. If Q is any transversal on DCosets (S_n, Cent (S_n, x), H), then $T = x^Q$ is a transversal on the orbits of ${}_HX$ (Colletti, 1999).

Given the information provided, the following example builds orbits on CC5 in S_8. The first step is to find the centralizers of any $x \in$ CC5, say $x = (1,5)(2,6)(3,7)(4,8)$. For notational purposes, < > and the permutations within < > represent a group via its generators. Using the Group, Algorithm, and Programming (GAP) software package:

$$\mathrm{Cent}\ (S_n, x) = <(4,8), (3,4)(7,8), (2,3)(6,7), (1,2)(5,6)>$$

and a transversal list of the DoubleCosets(S_n, Cent (S_n, x), H),

where $H = \{(), (5,6)(7,8), (5,7)(6,8), (5,8)(6,7)\}$ is:

Q = {( ), (7,8), (6,7), (6,7,8), (6,8,7), (6,8), (4,5), (4,5)(7,8), (4,5)(6,7,8),
(4,5,6), (4,5,6)(7,8), (4,5,6,7), (4,5,6,8,7), (4,5,7,8,6), (4,5,7)(6,8),
(3,4,5), (3,4,5)(7,8), (3,4,5)(6,7,8), (3,4,5,6), (3,4,5,6)(7,8),
(3,4,5,7,8,6), (3,5)(4,6), (3,5)(4,6)(7,8), (3,5,4,6), (3,5,4,6)(7,8),
(3,5)(4,7,6), (3,5,4,7,6), (2,3,4,5), (2,3,4,5)(7,8), (2,3,4,5)(6,7,8),
(2,3,5)(4,6), (2,3,5)(4,6)(7,8), (2,3,5)(4,7,6)}

Each permutation above, when conjugated with x, gives an element in each orbit of ${}_H$CC5, i.e., x^Q is an orbital transversal. Q is used to track orbits that have already been searched, i.e., once searched, an orbit's Q-element q is "checked off." The orbit is $(x^q)^H = x^{qH}$ and those of ${}_H$CC5 appear below, where each line is the orbit corresponding to a Q-element listed above.

[(1,5)(2,6)(3,7)(4,8), (1,6)(2,5)(3,8)(4,7), (1,7)(2,8)(3,5)(4,6), (1,8)(2,7)(3,6)(4,5)]
[(1,5)(2,6)(3,8)(4,7), (1,6)(2,5)(3,7)(4,8), (1,7)(2,8)(3,6)(4,5), (1,8)(2,7)(3,5)(4,6)]
[(1,5)(2,7)(3,6)(4,8), (1,6)(2,8)(3,5)(4,7), (1,7)(2,5)(3,8)(4,6), (1,8)(2,6)(3,7)(4,5)]
[(1,5)(2,7)(3,8)(4,6), (1,6)(2,8)(3,7)(4,5), (1,7)(2,5)(3,6)(4,8), (1,8)(2,6)(3,5)(4,7)]
[(1,5)(2,8)(3,6)(4,7), (1,6)(2,7)(3,5)(4,8), (1,7)(2,6)(3,8)(4,5), (1,8)(2,5)(3,7)(4,6)]
[(1,5)(2,8)(3,7)(4,6), (1,6)(2,7)(3,8)(4,5), (1,7)(2,6)(3,5)(4,8), (1,8)(2,5)(3,6)(4,7)]

[(1,4)(2,6)(3,7)(5,8), (1,4)(2,5)(3,8)(6,7), (1,4)(2,8)(3,5)(6,7), (1,4)(2,7)(3,6)(5,8)]
[(1,4)(2,6)(3,8)(5,7), (1,4)(2,5)(3,7)(6,8), (1,4)(2,8)(3,6)(5,7), (1,4)(2,7)(3,5)(6,8)]
[(1,4)(2,7)(3,8)(5,6), (1,4)(2,8)(3,7)(5,6), (1,4)(2,5)(3,6)(7,8), (1,4)(2,6)(3,5)(7,8)]
[(1,6)(2,4)(3,7)(5,8), (1,5)(2,4)(3,8)(6,7), (1,8)(2,4)(3,5)(6,7), (1,7)(2,4)(3,6)(5,8)]
[(1,6)(2,4)(3,8)(5,7), (1,5)(2,4)(3,7)(6,8), (1,8)(2,4)(3,6)(5,7), (1,7)(2,4)(3,5)(6,8)]
[(1,6)(2,7)(3,4)(5,8), (1,5)(2,8)(3,4)(6,7), (1,8)(2,5)(3,4)(6,7), (1,7)(2,6)(3,4)(5,8)]

[(1,6)(2,8)(3,4)(5,7), (1,5)(2,7)(3,4)(6,8), (1,8)(2,6)(3,4)(5,7), (1,7)(2,5)(3,4)(6,8)]
[(1,7)(2,4)(3,8)(5,6), (1,8)(2,4)(3,7)(5,6), (1,5)(2,4)(3,6)(7,8), (1,6)(2,4)(3,5)(7,8)]
[(1,7)(2,8)(3,4)(5,6), (1,8)(2,7)(3,4)(5,6), (1,5)(2,6)(3,4)(7,8), (1,6)(2,5)(3,4)(7,8)]
[(1,3)(2,6)(4,7)(5,8), (1,3)(2,5)(4,8)(6,7), (1,3)(2,8)(4,5)(6,7), (1,3)(2,7)(4,6)(5,8)]
[(1,3)(2,6)(4,8)(5,7), (1,3)(2,5)(4,7)(6,8), (1,3)(2,8)(4,6)(5,7), (1,3)(2,7)(4,5)(6,8)]
[(1,3)(2,7)(4,8)(5,6), (1,3)(2,8)(4,7)(5,6), (1,3)(2,5)(4,6)(7,8), (1,3)(2,6)(4,5)(7,8)]
[(1,6)(2,3)(4,7)(5,8), (1,5)(2,3)(4,8)(6,7), (1,8)(2,3)(4,5)(6,7), (1,7)(2,3)(4,6)(5,8)]
[(1,6)(2,3)(4,8)(5,7), (1,5)(2,3)(4,7)(6,8), (1,8)(2,3)(4,6)(5,7), (1,7)(2,3)(4,5)(6,8)]
[(1,7)(2,3)(4,8)(5,6), (1,8)(2,3)(4,7)(5,6), (1,5)(2,3)(4,6)(7,8), (1,6)(2,3)(4,5)(7,8)]
[(1,3)(2,4)(5,7)(6,8)]
[(1,3)(2,4)(5,8)(6,7)]
[(1,4)(2,3)(5,7)(6,8)]
[(1,4)(2,3)(5,8)(6,7)]
[(1,3)(2,4)(5,6)(7,8)]
[(1,4)(2,3)(5,6)(7,8)]
[(1,2)(3,6)(4,7)(5,8), (1,2)(3,5)(4,8)(6,7), (1,2)(3,8)(4,5)(6,7), (1,2)(3,7)(4,6)(5,8)]
[(1,2)(3,6)(4,8)(5,7), (1,2)(3,5)(4,7)(6,8), (1,2)(3,8)(4,6)(5,7), (1,2)(3,7)(4,5)(6,8)]
[(1,2)(3,7)(4,8)(5,6), (1,2)(3,8)(4,7)(5,6), (1,2)(3,5)(4,6)(7,8), (1,2)(3,6)(4,5)(7,8)]
[(1,2)(3,4)(5,7)(6,8)]
[(1,2)(3,4)(5,8)(6,7)]
[(1,2)(3,4)(5,6)(7,8)]

This method is simple and effective for small S_n, but as n grows, it becomes combinatorially burdensome to generate the transversal list. However, Section 3.4.1 did provide an example on partitioning a conjugacy class with orbits. Section 3.5 introduces a more effective method for traversing the solution space for larger S_n instances.

3.4.2 TDVRSP Specifications That Reduce Solution Space

So far, the solution space has reduced in size by taking advantage of TDVRSP cycle structure restrictions. As a result, only certain conjugacy classes are utilized in the solution space. Additionally, group actions partition these conjugacy classes. Another characteristic that further reduces solution space size is the letter combination restriction.

TDVRSP specification forbids assigning two or more vehicle letters to the same tour and assigning two or more customer letters to a tour without a vehicle letter in the tour. For example, if vehicles are lettered {1,2,3,4} and customers are lettered {5,6,7,8}, the following solution is infeasible because it has two vehicle letters in one tour.

(**1,2**)(3,5)(4,6)(7)(8)

Referring to the previous example of S_8 in Section 3.4.1, where CC5's orbits are listed, one notices that only six orbits have feasible letter combinations while the other 27 orbits list infeasible letter combinations. It is obvious that without imposing these restrictions, searching conjugacy classes and orbits littered with infeasible solutions is computationally inefficient.

The following is a method that further restricts the solution space by not allowing infeasible solutions from improper letter combinations. The method uses orbital transversals that maintain customer and vehicle letter integrity within the same cycle. This presumes the generating solution p maintains letter integrity. For example, using vehicle letters {1,2,3,4} and customer letters {5,6,7,8} on p = (1,5)(2,6)(3,7)(4,8), the following (partial) orbital transversal maintains feasibility.

$$Q = [(1,2), (6,7,8), (2,3)(7,8)]$$

$$p^Q = [(1,6)(2,5)(3,7)(4,8), (1,5)(2,8)(3,6)(4,7), (1,5)(2,8)(3,6)(4,7)]$$

Elements of orbital transversals that mix vehicle letters and customer letters within a cycle will violate feasibility, e.g.:

$$Q = [(4,6), (1,2,7,8), (1,6)]$$

$$p^Q = [(1,5)(2,4)(3,7)(6,8), (1,6)(2,3)(4,7)(5,8), (1,2)(3,7)(4,8)(5,6)]$$

This method will only work when p itself is feasible. Therefore, it is always important to maintain p with the correct letter combination restriction.

Using §3.3 notation, the above is a special instance of this general phenomenon: if $p \in X$ and $Q \subseteq \langle S(C), S(V) \rangle$, then $p^Q \subseteq X$. This simple statement embraces key efforts of this chapter and suggests move strategies that respect feasibility. That is, what are the "good" simplexes of $\langle S(C), S(V) \rangle$ that yield promising neighborhoods?

As a result, the solution space is further restricted in size. Instead of 105 possible solutions in CC5 of S_8, there are now only 24 possible solutions. The number of orbits within CC5 is now 6 instead of 33.

3.5 Orbital-planes

As mentioned in Section 3.4.1, traversing orbits within a conjugacy class generated by double coset transversals and centralizers are practical for small TDVRSP instances. However, this technique becomes computationally expensive as problem size increases. For the TDVRSP instances presented in this research, the method in Section 3.4.1 is impractical. Consequently, an approach that uses mutually exclusive groups to create orbits and orbital planes is developed for traversing the space of large problem instances. Orbital planes are defined in terms of S_n permutation structures as well as algebraically.

To describe orbital-planes using S_n cycle structure, define C and V as finite nonempty disjoint letter sets where $\{C_i\}_{i=1..n}$ partition C. Let $G_i = S(C_i)$ such that G_i is a symmetric group of letters C_i. Although V and C suggest "vehicles" and "customers" in the TDVRSP solution space, here we can simply treat sets as sets. The partition of C means C is the union of the disjoint nonempty C_i.

X is defined as all permutations in $S(V \cup C)$ whose nontrivial disjoint cyclic factors have a sole V-letter, where $\{X(k)\}$ partition X on cyclic form structure. The collection $\{X(k)\}$ represents the different cyclic form structures of X. For permutation $p \in X$, the symbol i-j denotes the cyclic-position in p of letter w iff w occupies the j'th position of the i'th nontrivial disjoint cyclic factor of p. For letter $w \notin \mathrm{mov}(p)$, the null symbol 0-0 is warranted. This presumes a convention that unambiguously defines the

ordinal sequence of a permutation's nontrivial disjoint cyclic factors and the letter positions within each. As an instance of the latter, suppose π is a *p*-factor and *v* is π's *V*-letter. Then letter $c \in C$ is in the *n*'th position of π iff $c = (v)\pi^{n-1}$. For $p \in X$, $N_i(p)$ denotes the set of all non-null cyclic-positions in p of all C_i -letters.

Thus, using notation from the previous paragraph, the definition of orbital planes follow:

Definition (*Orbital planes {X(k,m)}*), are defined as the partitioning of *X(k)* according to the following equivalence relation on *X(k)*: $p \sim q$ iff $N_i(p) = N_i(q)\ \forall\ i$. Where *X* are all permutations in $S(V \cup C)$ whose nontrivial disjoint cyclic factors have a sole *V*-letter, the set {*X(k)*} partition *X* on cyclic form structure. For $p \in X$, $N_i(p)$ denotes the set of all non-null cyclic-positions in p of all C_i -letters where $\{C_i\}$ $i = 1..n$ partition *C* as finite nonempty disjoint letter sets. The orbital plane index is represented by m.

*Assumes dictionary ordering and $|V|<|C|$.

Algebraically, orbital planes are defined in much simpler notation, thus indicating the power of group theory as a mechanism within tabu search. In terms of group theory, the orbital plane of *p* is defined below.

Definition (*orbital plane of p*) is p^G, where $G = \prod_i S(C_i)$.

The following describes an orbital plane using an example from S_{14}.

Let $p = (1,8,14,11)(2,6)(3,7,10)(4,9)(5,12,13)$
$V = \{1,2,3,4,5\}$,
$C = \{6,7,8,9,10,11,12,13,14\}$
where
$C_1 = \{6,7,14\}$, $C_2 = \{8,9,10\}$, and $C_3 = \{11,12,13\}$ partition *C*.

The set X, which contains all the possible cyclic form structures of S_n, has an assigned $X(k)$ representing the cyclic form structure (x,x,x,x)(x,x)(x,x, x)(x,x)(x,x,x) from conjugacy class $4^1 3^2 2^2$.

Letters $c \in C_i \ \forall i$ each have a position within p. For example, letter 14 is in the 1st cycle, 3rd position, represented as position 1-3 in p. Letter 6 is in the 2nd cycle, 2nd position, and is represented as position 2-2 in p. Letter 7 is represented as position 3-2 in p. Consequently, the set of letters in C_1 occupy positions {1-3,2-2, 3-2}. The letters in C_2 occupy positions {1-2,3-3,4-2} and the letters in C_3 occupy positions {1-4,5-2,5-3}. $\{N_i(p)| \ \forall i\}$ is the unique collection of positions for C_1, C_2, and C_3 , represented as {{1-3,2-2, 3-2}, {1-2,3-3,4-2}, {1-4,5-2,5-3}}. There are a total of $|C|$ possible positions in the cyclic form structure (in this case, 9 positions).

The orbital plane consists of any permutation q in which the letters of C_1 occupy positions {1-3,2-2, 3-2}, the letters of C_2 occupy positions {1-2,3-3,4-2}, and the letters of C_3 occupy positions {1-4,5-2,5-3}. The letters of C_i may occupy any position within its respective position grouping.

Consequently, movements of letters within a C_i keep q in the same orbital plane, but movements of letters between C_i's move q to a different orbital plane. For example, the current orbital plane is represented as:

C_1 letters occupy positions {1-3,**2-2**,3-2}
C_2 letters occupy positions {**1-2**,3-3,4-2}
C_3 letters occupy positions {1-4,5-2,5-3}

If a letter is swapped between C_1 and C_2 that occupy positions 2-2 and 1-2 respectively, then a new orbital plane is formed and represented as:

C_1 letters occupy positions {1-3,**1-2**,3-2}
C_2 letters occupy positions {**2-2**,3-3,4-2}
C_3 letters occupy positions {1-4,5-2,5-3}

The number of orbital planes in $X(k)$ is equal to the number of positioning combinations and can be calculated using ${}_nC_r = \dfrac{n!}{r!(n-r)!}$. For this example, C_1 has ${}_9C_3$ = 84 position groups, C_2 has ${}_6C_3$ = 20 position groups given C_1, and C_3 has ${}_3C_3$ =1 position group given C_1 and C_2. Therefore, the total number of positioning combinations is 84*20*1=1680 and the total number of orbital planes in $X(k)$ is 1680.

Orbital planes are unique for each cyclic form structure within a conjugacy class. As the cyclic form structure changes, the positions change. For example, if the new cyclic form structure is (x,x,x)(x,x)(x,x,x,x)(x,x)(x,x,x) from conjugacy class $4^1 3^2 2^2$ instead of (x,x,x,x)(x,x)(x,x,x)(x,x)(x,x,x) from conjugacy class $4^1 3^2 2^2$, then position 1-4 leaves and position 3-4 enters. The new position provides for 1680 new combinations for C_1, C_2, and C_3. Therefore, the number of orbital planes per conjugacy class, using previously defined notation is

$$= \sum_{k=1}^{m} {}_nC_{r(k)} \quad \text{where } m=|X_k| \text{ and } r(k) \text{ represents } |C| \ \forall\ k$$

Orbits exclusively and exhaustively partition orbital planes and are created by using mutually exclusive groups on $p \in X$. Let G represent a disjoint set of groups $\{G_i : i = 1,2,\ldots,r\}$ where G_i is a defined group derived from letters in C_i. Since a group G_i only moves letters C_i within p, $c \in C_i$ in q maintains $N_i(p) = N_i(q)$.

In order to further demonstrate orbits and orbital planes, we use $X_{i,j,k}$ as the positioning and ordering of C_i $\forall$ i. Each i indexes the C_i and $i = 1,2,\ldots,r$. Each j represents an instance of $N_i(p)$ where $j = 1,2,\ldots,s$. Each k is an ordering of letters within a specified j, where $k = 1,2,\ldots,t$ and $K \equiv \{k = 1,2,\ldots,t\}$. Index s is dependent on the cyclic form structure and index t is dependent on the size of the group action.

A group G_i acting on p creates an orbit. Within that orbit exists a collection of solutions represented as $x_{1,j,k}$, $x_{2,j,k},\ldots,x_{r,j,k}$ per solution where j cannot have duplicate values for any $i = 1,2,\ldots,r$ within p. Index j is a specified collection of positions within p, and k = the (C_i) letter ordering from group elements $1,2,\ldots,t$ in G_i. There exist up to t solutions in Orbit (G_i, p). A unique orbital plane exists for each combination of j and a unique orbit exists for each combination of k within an orbital plane. A unique solution exists for each unique i, j, and k.

For example, an orbit is represented by $X_{1,1,K}, x_{2,2,1}, x_{3,3,1}, \ldots, x_{r,4,1}$, where each x_i has a specified positioning j and ordering k of letters C_i. $X_{1,1,K}$ are the different ordering sequences of the C_1 letters in the collection of positions 1. An orbit, where |Orbit (G_i, p)| = 3, is depicted in Figure 3.5.

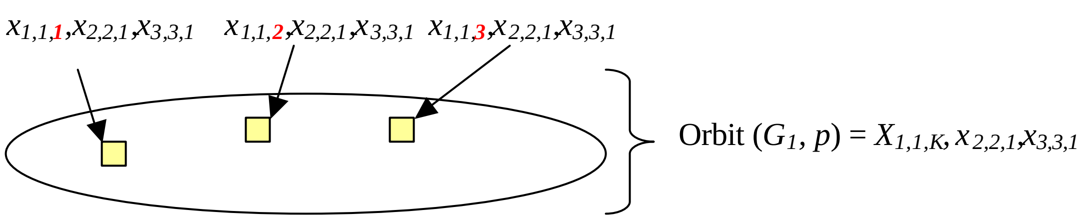

Figure 3.5 Orbit Example

Two orbital planes are portrayed in Figure 3.6. Each orbital plane is defined by $N_i(p)$ for the C_i sets. Note that combinations of k define each orbit within a plane. For

example, across row 1 of the top orbital plane exist the index k combinations (K,1,1), (K,2,1), (K,3,1) , and (K,4,1). The second row is composed of index k combinations (1,K,1), (2,K,1), and (3,K,1). The bottom orbital plane differs from the top orbital plane based on the difference of index j for each x. For example, the top plane has the combination 1, 2, 3 for the j indices, whereas the lower plane has combination 4, 5, 3 for the j indices. This is caused by a swap of letters between C_1 and C_2, causing a change in positioning sequence.

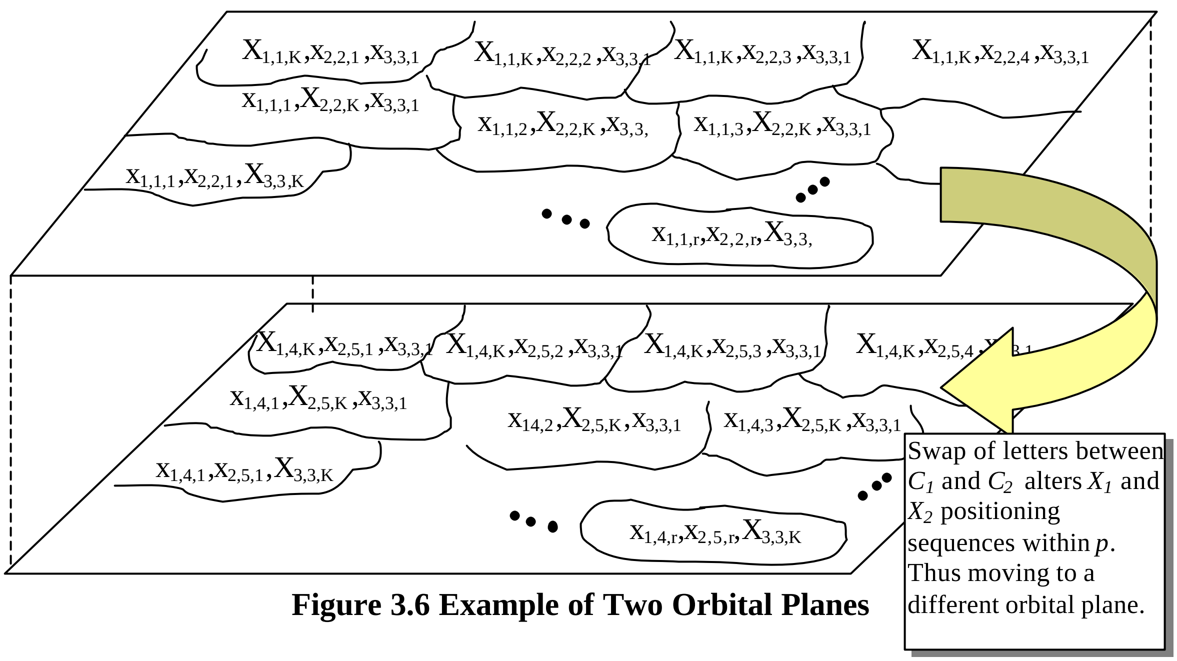

Figure 3.6 Example of Two Orbital Planes

A method to move from one plane to another is performed by moving letters between different C_i sets in p. For $p \in X$, suppose $c \in S(Ci \cup Cj)$ maps some p-letter in C_i into a p-letter in C_j. Then c represents an interplane move, i.e., p and p^c are in different orbital planes.

Orbital planes provide a more granular partition of the solution space than cyclic form structures and conjugacy classes, and as a result, orbital traversal lists decrease in

size from those generated in Section 3.4.1. In Section 3.4.1, a method that creates orbital transversal lists within a conjugacy class was presented. As mentioned in that section, generating these traversal lists becomes combinatorially burdensome as S_n increases in size. Fortunately, given the decrease in solution space size of an orbital plane, methods that generate orbital traversal lists become less computationally burdensome. The following are three methods to traverse the orbital plane solution space. They include the *Exclusive and Exhaustive Method*, *Exhaustive Method*, and *Greedy Method*.

3.5.1 Exclusive and Exhaustive Orbital Plane Traversal Method

In this Section, orbital planes are traversed using orbits that exclusively and exhaustively partition the orbital plane. This method is simple, but can be computationally burdensome for very large problem sizes. However, this method is less computationally burdensome than the method presented in Section 3.4.1.

As previously defined, let $G_i = S(C_i)$, $i = 1,\ldots,n$. For any $p \in X$, we will partition its orbital plane using the algorithm below.

1. Select an arbitrary G_k.
2. Build $H = \prod_{i \neq k} G_i$ (H is a group)
3. For each $q \in p^H$ (ie an orbit of ${}_HX$) evaluate Orbit (G_k, q)

The following is an example of this method.

G_1 = <(10,11,12), (10,11)> = $S(C_1)$
G_2 = <(13,14,15), (13,14)> = $S(C_2)$
G_3 = <(16,17,18), (16,17)> = $S(C_3)$
p = (1,10)(2,11)(3,12)(4,13)(5,14)(6,15)(7,16)(8,17)(9,18)

Step 1:
Select G_3 .
Step 2:
$H = G_1 * G_2$;

Step 3:

Let $T = p^H$

$T =$ [(1,10)(2,11)(3,12)(4,13)(5,14)(6,15)(7,16)(8,17)(9,18),
(1,11)(2,12)(3,10)(4,13)(5,14)(6,15)(7,16)(8,17)(9,18),
(1,11)(2,10)(3,12)(4,13)(5,14)(6,15)(7,16)(8,17)(9,18),
(1,10)(2,11)(3,12)(4,14)(5,15)(6,13)(7,16)(8,17)(9,18),
(1,10)(2,11)(3,12)(4,14)(5,13)(6,15)(7,16)(8,17)(9,18),
(1,12)(2,10)(3,11)(4,13)(5,14)(6,15)(7,16)(8,17)(9,18),
(1,10)(2,12)(3,11)(4,13)(5,14)(6,15)(7,16)(8,17)(9,18),
(1,11)(2,12)(3,10)(4,14)(5,15)(6,13)(7,16)(8,17)(9,18),
(1,11)(2,12)(3,10)(4,14)(5,13)(6,15)(7,16)(8,17)(9,18),
(1,12)(2,11)(3,10)(4,13)(5,14)(6,15)(7,16)(8,17)(9,18),
(1,11)(2,10)(3,12)(4,14)(5,15)(6,13)(7,16)(8,17)(9,18),
(1,11)(2,10)(3,12)(4,14)(5,13)(6,15)(7,16)(8,17)(9,18),
(1,10)(2,11)(3,12)(4,15)(5,13)(6,14)(7,16)(8,17)(9,18),
(1,10)(2,11)(3,12)(4,13)(5,15)(6,14)(7,16)(8,17)(9,18),
(1,10)(2,11)(3,12)(4,15)(5,14)(6,13)(7,16)(8,17)(9,18),
(1,12)(2,10)(3,11)(4,14)(5,15)(6,13)(7,16)(8,17)(9,18),
(1,12)(2,10)(3,11)(4,14)(5,13)(6,15)(7,16)(8,17)(9,18),
(1,10)(2,12)(3,11)(4,14)(5,15)(6,13)(7,16)(8,17)(9,18),
(1,10)(2,12)(3,11)(4,14)(5,13)(6,15)(7,16)(8,17)(9,18),
(1,11)(2,12)(3,10)(4,15)(5,13)(6,14)(7,16)(8,17)(9,18),
(1,11)(2,12)(3,10)(4,13)(5,15)(6,14)(7,16)(8,17)(9,18),
(1,11)(2,12)(3,10)(4,15)(5,14)(6,13)(7,16)(8,17)(9,18),
(1,12)(2,11)(3,10)(4,14)(5,15)(6,13)(7,16)(8,17)(9,18),
(1,12)(2,11)(3,10)(4,14)(5,13)(6,15)(7,16)(8,17)(9,18),
(1,11)(2,10)(3,12)(4,15)(5,13)(6,14)(7,16)(8,17)(9,18),
(1,11)(2,10)(3,12)(4,13)(5,15)(6,14)(7,16)(8,17)(9,18),
(1,11)(2,10)(3,12)(4,15)(5,14)(6,13)(7,16)(8,17)(9,18),
(1,12)(2,10)(3,11)(4,15)(5,13)(6,14)(7,16)(8,17)(9,18),
(1,12)(2,10)(3,11)(4,13)(5,15)(6,14)(7,16)(8,17)(9,18),
(1,12)(2,10)(3,11)(4,15)(5,14)(6,13)(7,16)(8,17)(9,18),
(1,10)(2,12)(3,11)(4,15)(5,13)(6,14)(7,16)(8,17)(9,18),
(1,10)(2,12)(3,11)(4,13)(5,15)(6,14)(7,16)(8,17)(9,18),
(1,10)(2,12)(3,11)(4,15)(5,14)(6,13)(7,16)(8,17)(9,18),
(1,12)(2,11)(3,10)(4,15)(5,13)(6,14)(7,16)(8,17)(9,18),
(1,12)(2,11)(3,10)(4,13)(5,15)(6,14)(7,16)(8,17)(9,18),
(1,12)(2,11)(3,10)(4,15)(5,14)(6,13)(7,16)(8,17)(9,18)]

Orbital traversal example (lists orbits 1,2,…,36 of the orbital plane and their elements)

O_1 = Orbit (G_3, t_1);
[(1,10)(2,11)(3,12)(4,13)(5,14)(6,15)(7,16)(8,17)(9,18),
(1,10)(2,11)(3,12)(4,13)(5,14)(6,15)(7,17)(8,18)(9,16),
(1,10)(2,11)(3,12)(4,13)(5,14)(6,15)(7,17)(8,16)(9,18),

(1,10)(2,11)(3,12)(4,13)(5,14)(6,15)(7,18)(8,16)(9,17),
(1,10)(2,11)(3,12)(4,13)(5,14)(6,15)(7,16)(8,18)(9,17),
(1,10)(2,11)(3,12)(4,13)(5,14)(6,15)(7,18)(8,17)(9,16)]

O_2 = Orbit (G_3, t_2);
[(1,11)(2,12)(3,10)(4,13)(5,14)(6,15)(7,16)(8,17)(9,18),
(1,11)(2,12)(3,10)(4,13)(5,14)(6,15)(7,17)(8,18)(9,16),
(1,11)(2,12)(3,10)(4,13)(5,14)(6,15)(7,17)(8,16)(9,18),
(1,11)(2,12)(3,10)(4,13)(5,14)(6,15)(7,18)(8,16)(9,17),
(1,11)(2,12)(3,10)(4,13)(5,14)(6,15)(7,16)(8,18)(9,17),
(1,11)(2,12)(3,10)(4,13)(5,14)(6,15)(7,18)(8,17)(9,16)]

…………

O_{36} = Orbit (G_3, t_{36});
[(1,12)(2,11)(3,10)(4,15)(5,14)(6,13)(7,16)(8,17)(9,18),
(1,12)(2,11)(3,10)(4,15)(5,14)(6,13)(7,17)(8,18)(9,16),
(1,12)(2,11)(3,10)(4,15)(5,14)(6,13)(7,17)(8,16)(9,18),
(1,12)(2,11)(3,10)(4,15)(5,14)(6,13)(7,18)(8,16)(9,17),
(1,12)(2,11)(3,10)(4,15)(5,14)(6,13)(7,16)(8,18)(9,17),
(1,12)(2,11)(3,10)(4,15)(5,14)(6,13)(7,18)(8,17)(9,16)]

The advantage of this method is the ability to exhaustively search the orbital plane without evaluating redundant solutions. The method also prevents cycling within the search. Consequently, the disadvantage of this method is the computational burden of an exhaustive search, especially if this portion of the solution space does not produce high quality solutions. Another disadvantage is the computational burden of generating the traversal list. As the number and size of G_i increases, so does the size of the traversal lists. This computation must be performed before a single orbit is created or evaluated.

3.5.2 Exhaustive Orbital Plane Traversal Method

In this Section, orbital planes are traversed using orbits that exhaustively partition the orbital plane. This method is similar to that of Section 3.5.1, but eliminates one of the

previous method's disadvantages by allowing the generation and evaluation of orbits as the algorithm progressed.

As previously defined, let G_i = S(C_i), i = 1,…,n. For any $p \in X$, we will partition its orbital plane using the algorithm below. The following algorithm exhaustively traverses the orbital plane solution space using a method that successively generates orbits with mutually exclusive groups.

a. Generate Orbit (G_1,p).
b. Let T_1 = {elements∈ Orbit (G_1,p)} = p^{G1}
c. Generate Orbit (G_2,t) $\forall t \in T_1$
d. Let T_2 = {elements∈ {Orbit (G_2,t_1), Orbit (G_2,t_2),…,Orbit (G_2,t_m)}}= p^{G1*G2}
e. Continue sequence until $i = n$, where $T_i = p^{G1*G2*\ldots*Gi}$

The algorithm results in the generation of a collection of orbits that exhaustively partition the solution space. Figure 3.7 conceptually portrays the methodology.

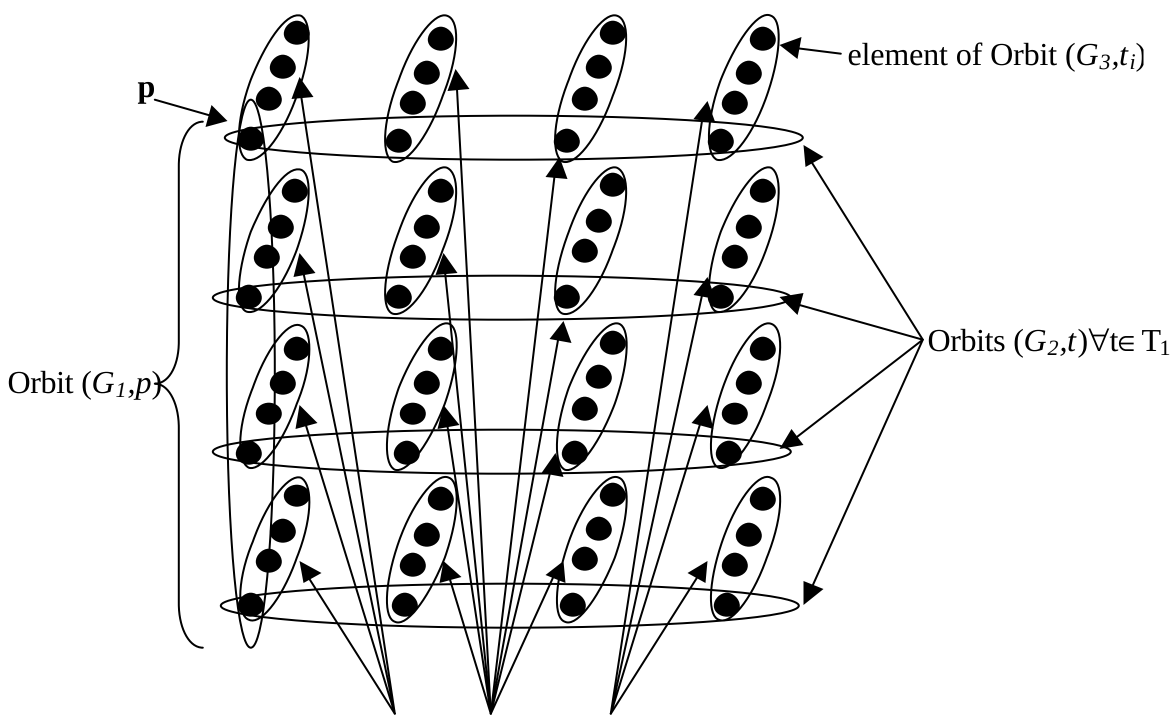

Figure 3.7 Orbital Plane Partitioning Through Successive Group Actions

The following is an example of this method, where 3 groups generate the orbital plane.

G_1 = <(10,11,12), (10,11)> = S(C_1)
G_2 = <(13,14,15), (13,14)> = S(C_2)
G_3 = <(16,17,18), (16,17)> = S(C_3)
p = (1,10)(2,11)(3,12)(4,13)(5,14)(6,15)(7,16)(8,17)(9,18)

Step 1&2:

T_1 = {elements of Orbit O_1 }
O_1 = Orbit (*G1*, *p*);
[(1,10)(2,11)(3,12)(4,13)(5,14)(6,15)(7,16)(8,17)(9,18),
(1,11)(2,12)(3,10)(4,13)(5,14)(6,15)(7,16)(8,17)(9,18),
(1,11)(2,10)(3,12)(4,13)(5,14)(6,15)(7,16)(8,17)(9,18),
(1,12)(2,10)(3,11)(4,13)(5,14)(6,15)(7,16)(8,17)(9,18),
(1,10)(2,12)(3,11)(4,13)(5,14)(6,15)(7,16)(8,17)(9,18),
(1,12)(2,11)(3,10)(4,13)(5,14)(6,15)(7,16)(8,17)(9,18)]

Step 3&4:

T_2 = {elements of Orbits O_2 through O_7}
O_2 = Orbit (G_2, T_1[1]);
[(1,10)(2,11)(3,12)(4,13)(5,14)(6,15)(7,16)(8,17)(9,18),
(1,10)(2,11)(3,12)(4,14)(5,15)(6,13)(7,16)(8,17)(9,18),
(1,10)(2,11)(3,12)(4,14)(5,13)(6,15)(7,16)(8,17)(9,18),
(1,10)(2,11)(3,12)(4,15)(5,13)(6,14)(7,16)(8,17)(9,18),
(1,10)(2,11)(3,12)(4,13)(5,15)(6,14)(7,16)(8,17)(9,18),
(1,10)(2,11)(3,12)(4,15)(5,14)(6,13)(7,16)(8,17)(9,18)]

O_3 = Orbit (G_2, T_1[2]);
[(1,11)(2,12)(3,10)(4,13)(5,14)(6,15)(7,16)(8,17)(9,18),
(1,11)(2,12)(3,10)(4,14)(5,15)(6,13)(7,16)(8,17)(9,18),
(1,11)(2,12)(3,10)(4,14)(5,13)(6,15)(7,16)(8,17)(9,18),
(1,11)(2,12)(3,10)(4,15)(5,13)(6,14)(7,16)(8,17)(9,18),
(1,11)(2,12)(3,10)(4,13)(5,15)(6,14)(7,16)(8,17)(9,18),
(1,11)(2,12)(3,10)(4,15)(5,14)(6,13)(7,16)(8,17)(9,18)]

O_4 = Orbit (G_2, T_1[3]);
[(1,11)(2,10)(3,12)(4,13)(5,14)(6,15)(7,16)(8,17)(9,18),
(1,11)(2,10)(3,12)(4,14)(5,15)(6,13)(7,16)(8,17)(9,18),
(1,11)(2,10)(3,12)(4,14)(5,13)(6,15)(7,16)(8,17)(9,18),
(1,11)(2,10)(3,12)(4,15)(5,13)(6,14)(7,16)(8,17)(9,18),
(1,11)(2,10)(3,12)(4,13)(5,15)(6,14)(7,16)(8,17)(9,18),
(1,11)(2,10)(3,12)(4,15)(5,14)(6,13)(7,16)(8,17)(9,18)]

O_5 = Orbit (G_2, T_1[4]);

[(1,12)(2,10)(3,11)(4,13)(5,14)(6,15)(7,16)(8,17)(9,18),
(1,12)(2,10)(3,11)(4,14)(5,15)(6,13)(7,16)(8,17)(9,18),
(1,12)(2,10)(3,11)(4,14)(5,13)(6,15)(7,16)(8,17)(9,18),
(1,12)(2,10)(3,11)(4,15)(5,13)(6,14)(7,16)(8,17)(9,18),
(1,12)(2,10)(3,11)(4,13)(5,15)(6,14)(7,16)(8,17)(9,18),
(1,12)(2,10)(3,11)(4,15)(5,14)(6,13)(7,16)(8,17)(9,18)]

O_6 = Orbit (G_2, T_1[5]);

[(1,10)(2,12)(3,11)(4,13)(5,14)(6,15)(7,16)(8,17)(9,18),
(1,10)(2,12)(3,11)(4,14)(5,15)(6,13)(7,16)(8,17)(9,18),
(1,10)(2,12)(3,11)(4,14)(5,13)(6,15)(7,16)(8,17)(9,18),
(1,10)(2,12)(3,11)(4,15)(5,13)(6,14)(7,16)(8,17)(9,18),
(1,10)(2,12)(3,11)(4,13)(5,15)(6,14)(7,16)(8,17)(9,18),
(1,10)(2,12)(3,11)(4,15)(5,14)(6,13)(7,16)(8,17)(9,18)]

O_7 = Orbit (G_2, T_1[6]);

[(1,12)(2,11)(3,10)(4,13)(5,14)(6,15)(7,16)(8,17)(9,18),
(1,12)(2,11)(3,10)(4,14)(5,15)(6,13)(7,16)(8,17)(9,18),
(1,12)(2,11)(3,10)(4,14)(5,13)(6,15)(7,16)(8,17)(9,18),
(1,12)(2,11)(3,10)(4,15)(5,13)(6,14)(7,16)(8,17)(9,18),
(1,12)(2,11)(3,10)(4,13)(5,15)(6,14)(7,16)(8,17)(9,18),
(1,12)(2,11)(3,10)(4,15)(5,14)(6,13)(7,16)(8,17)(9,18)]

Step 5:

O_8 = Orbit (G_3, T_2[1]);

[(1,11)(2,12)(3,10)(4,13)(5,14)(6,15)(7,16)(8,17)(9,18),
(1,11)(2,12)(3,10)(4,13)(5,14)(6,15)(7,17)(8,18)(9,16),
(1,11)(2,12)(3,10)(4,13)(5,14)(6,15)(7,17)(8,16)(9,18),
(1,11)(2,12)(3,10)(4,13)(5,14)(6,15)(7,18)(8,16)(9,17),
(1,11)(2,12)(3,10)(4,13)(5,14)(6,15)(7,16)(8,18)(9,17),
(1,11)(2,12)(3,10)(4,13)(5,14)(6,15)(7,18)(8,17)(9,16)]

O_9 = Orbit (G_3, T_2[2]);

[(1,11)(2,12)(3,10)(4,14)(5,15)(6,13)(7,16)(8,17)(9,18),
(1,11)(2,12)(3,10)(4,14)(5,15)(6,13)(7,17)(8,18)(9,16),
(1,11)(2,12)(3,10)(4,14)(5,15)(6,13)(7,17)(8,16)(9,18),
(1,11)(2,12)(3,10)(4,14)(5,15)(6,13)(7,18)(8,16)(9,17),
(1,11)(2,12)(3,10)(4,14)(5,15)(6,13)(7,16)(8,18)(9,17),
(1,11)(2,12)(3,10)(4,14)(5,15)(6,13)(7,18)(8,17)(9,16)]

………….

O_{43} = Orbit (H_3, T_2[36]);
[(1,12)(2,11)(3,10)(4,13)(5,14)(6,15)(7,16)(8,17)(9,18),
(1,12)(2,11)(3,10)(4,14)(5,15)(6,13)(7,17)(8,18)(9,16),
(1,12)(2,11)(3,10)(4,14)(5,13)(6,15)(7,17)(8,16)(9,18),
(1,12)(2,11)(3,10)(4,15)(5,13)(6,14)(7,18)(8,16)(9,17),
(1,12)(2,11)(3,10)(4,13)(5,15)(6,14)(7,16)(8,18)(9,17),
(1,12)(2,11)(3,10)(4,15)(5,14)(6,13)(7,18)(8,17)(9,16)]

In this example, there were a total of 43 orbits generated. One orbit was generated by G_1, the next six orbits were generated by G_2, and the final 36 orbits were generated by G_3. The 36 orbits generated by G_3 exclusively and exhaustively partition the orbital plane; however, the first seven orbits are contained within the last 36 orbits. Though not any cause for alarm, it is important to recognize there exist redundant solutions generated using this method. Therefore, all the orbits generated in this method are not mutually exclusive. A method to alleviate redundancy is to not include the identity element within the G_i groups.

The advantage of using this method is the ability to generate and evaluate orbits as the algorithm proceeds. This allows immediate feedback on the quality of solutions within the orbital plane solution space. The disadvantage is the existence of permutations that reside in multiple orbits. This causes some computational inefficiency if those permutations are evaluated multiple times.

3.5.3 Greedy Orbital Plane Traversal Method

The greedy orbital plane traversal method is a quick and easy method for traversing portions of an orbital plane. The method is not an exhaustive search of the solution space and is based on generating orbits from permutations with good objective function values. The generated *p*-vector trends towards orbital plane space that provides quality objective function values.

As previously defined, let $G_i = S(C_i)$, $i = 1,\ldots,n$ and let $p \in X$. The following algorithm greedily traverses the orbital plane solution space.

1. Explore the orbital plane of a given p.
2. for $i = 1,2,\ldots,n$
   a. generate Orbit (G_i, p)
   b. pick best q from Orbit(G_i, p) and set $p = q$.
3. Repeat step 2 until a user defined exit point is met

The following is an example of this method, where 3 groups generate the orbital plane.

G_1 = <(10,11,12), (10,11)> = S(C_1)
G_2 = <(13,14,15), (13,14)> = S(C_2)
G_3 = <(16,17,18), (16,17)> = S(C_3)

Step 1: p = (1,10)(2,11)(3,12)(4,13)(5,14)(6,15)(7,16)(8,17)(9,18)

Step 2a: O_1 = Orbit (G_1, p);
[(1,10)(2,11)(3,12)(4,13)(5,14)(6,15)(7,16)(8,17)(9,18),
(1,11)(2,12)(3,10)(4,13)(5,14)(6,15)(7,16)(8,17)(9,18),
(1,11)(2,10)(3,12)(4,13)(5,14)(6,15)(7,16)(8,17)(9,18),
(1,12)(2,10)(3,11)(4,13)(5,14)(6,15)(7,16)(8,17)(9,18),
(1,10)(2,12)(3,11)(4,13)(5,14)(6,15)(7,16)(8,17)(9,18),
(1,12)(2,11)(3,10)(4,13)(5,14)(6,15)(7,16)(8,17)(9,18)]

Step 2b: p = (1,11)(2,12)(3,10)(4,13)(5,14)(6,15)(7,16)(8,17)(9,18),

Step 2a: O_2 = Orbit (G_2, p);
[(1,11)(2,12)(3,10)(4,13)(5,14)(6,15)(7,16)(8,17)(9,18),
(1,11)(2,12)(3,10)(4,14)(5,15)(6,13)(7,16)(8,17)(9,18),
(1,11)(2,12)(3,10)(4,14)(5,13)(6,15)(7,16)(8,17)(9,18),
(1,11)(2,12)(3,10)(4,15)(5,13)(6,14)(7,16)(8,17)(9,18),
(1,11)(2,12)(3,10)(4,13)(5,15)(6,14)(7,16)(8,17)(9,18),
(1,11)(2,12)(3,10)(4,15)(5,14)(6,13)(7,16)(8,17)(9,18)]

Step 2b: p = (1,11)(2,12)(3,10)(4,15)(5,14)(6,13)(7,16)(8,17)(9,18)]

Step 2a: O_3 = Orbit (G_3, p);

[(1,11)(2,12)(3,10) 4,15)(5,14)(6,13)(7,16)(8,17)(9,18),
(1,11)(2,12)(3,10)(4,15)(5,14)(6,13)(7,17)(8,18)(9,16),
(1,11)(2,12)(3,10)(4,15)(5,14)(6,13)(7,17)(8,16)(9,18),
(1,11)(2,12)(3,10)(4,15)(5,14)(6,13)(7,18)(8,16)(9,17),
(1,11)(2,12)(3,10)(4,15)(5,14)(6,13)(7,16)(8,18)(9,17),
(1,11)(2,12)(3,10)(4,15)(5,14)(6,13)(7,18)(8,17)(9,16)]

……………………..

The greedy orbital plane traversal method has its advantages and disadvantages. The method is advantageous because it has the ability to find better solutions more quickly than the previous methods. The search proceeds in a greedy direction towards good solutions in the orbital plane. Although it is greedy, the search will not get trapped in local optimal space because the method does not require p be an improving solution. The method only requires p be the best solution within the evaluated orbit. This is the best method of the three presented for traversing very large solutions spaces. The method also has disadvant ages. First, the search is not exhaustive. If a great solution exists within the orbital plane, this method does not guarantee finding it. Second, the orbits are not mutually exclusive. It is possible for two orbits generated by different p^{Gi} to share one permutation. Lastly, it is possible the search may cycle. In order to prevent cycling, a tabu list is implemented for orbits.

Given the size of the TDVRSP instances, the greedy orbital plane traversal method is implemented as the method of choice for the GTTS. This method allows the search to quickly find improving solutions. The GTTS makes up for the method's disadvantages by implementing an orbit tabu list and allowing the search to proceed into previously visited orbital planes in order to find the very best solutions.

3.5.4 Summary

In summary, orbital planes are defined as the collection of positions for S_n letter subsets within a permutation's cyclic form structure. In algebraic terms, *p*'s orbital plane is p^G s.t. $G = \prod_i S(C_i)$. Orbital planes are derived through the use of mutually exclusive groups acting on *p*, in which the group $G_i = S(C_i)$. Orbits exclusively and exhaustively partition orbital planes.

The orbits, orbital planes, cyclic form structures, and conjugacy classes presented thus far provide means to partition the solution space. The next section describes the solution space partition hierarchy.

3.6 Solution Space Partition Hierarchy

One of the advantages of group theory within a tabu search framework is the ability to partition the solution space. This provides the search more intelligent information on where it has been and where it should proceed. For example, if elite solutions are found within a specific orbital plane or cyclic form structure or conjugacy class, then the search conducts intensification within that partition of the space as directed.

This section utilizes the group theory concepts presented thus far and provides a solution space hierarchy that is used within the GTTS. The solution space hierarchy consists of conjugacy classes, cyclic form structures, orbital planes, and orbits.

The first order partition is the conjugacy class. Conjugacy classes, introduced in Section 3.2, partition S_n's solution space based on the solution's cyclic factors. For example, solutions with the cyclic form structures (x,x,x)(x,x)(x,x,x)(x,x) and (x,x,x)(x,x,x)(x,x)(x,x) are in the same conjugacy class and are described by factors $3^2 2^2$.

Conjugacy classes of any group are mutually exclusive and exhaustive (Colletti, 1999). Figure 3.8 is an illustration of conjugacy classes that partition S_7 with 2,3 and 4 cycle factor combinations. Unit cycles are ignored.

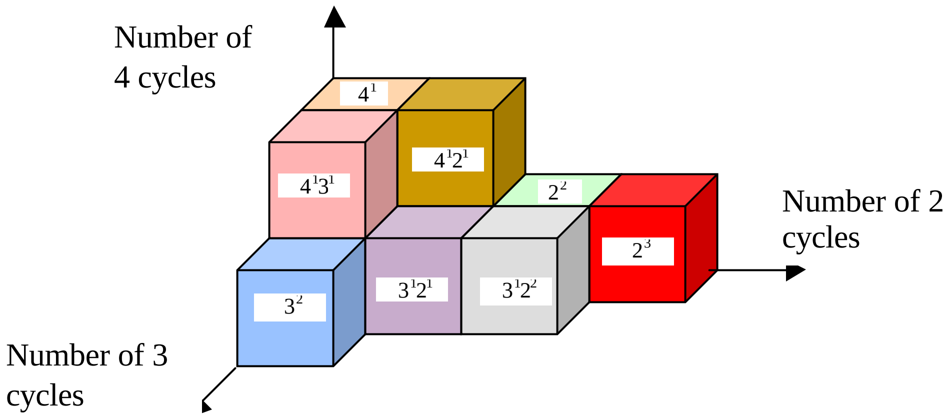

Figure 3.8 Conjugacy Class Partitions For S_7

The second order partitions are the cyclic form structures within a conjugacy class. Although group theory does not have a mechanism for this partition, we can do so logically. Cyclic form structures are specified by the unique ordering of the permutation cycles. For example, a conjugacy class of S_{12} represented by factors $3^2 2^3 1^0$ has the following cyclic form structures.

3-3-2-2-2	(x,x,x)(x,x,x)(x,x)(x,x)(x,x)
3-2-3-2-2	(x,x,x)(x,x)(x,x,x)(x,x)(x,x)
3-2-2-3-2	(x,x,x)(x,x)(x,x)(x,x,x)(x,x)
3-2-2-2-3	(x,x,x)(x,x)(x,x)(x,x)(x,x,x)
2-3-3-2-2	(x,x)(x,x,x)(x,x,x)(x,x)(x,x)
2-3-2-3-2	(x,x)(x,x,x)(x,x)(x,x,x)(x,x)
2-3-2-2-3	(x,x)(x,x,x)(x,x)(x,x)(x,x,x)
2-2-3-3-2	(x,x)(x,x)(x,x,x)(x,x,x)(x,x)
2-2-3-2-3	(x,x)(x,x)(x,x,x)(x,x)(x,x,x)
2-2-2-3-3	(x,x)(x,x)(x,x)(x,x,x)(x,x,x)

The number of cyclic form structures per conjugacy class is represented by the following permutation equation. The total number of cycles is represented by *n* and the number of each cycle type is represented by n_i, where $n = \sum n_i$ for $i = 1,\ldots,k$.

$$nP\ n_1, n_2, \ldots n_k = \frac{n!}{n_1!, n_2!, \ldots, n_k!}$$

For conjugacy class $3^2 2^3 1^0$, the total number of cyclic form structures is $\frac{5!}{2!3!0!} = 10$

Cyclic form structures exclusively and exhaustively partition the conjugacy classes and are graphically displayed in Figure 3.9.

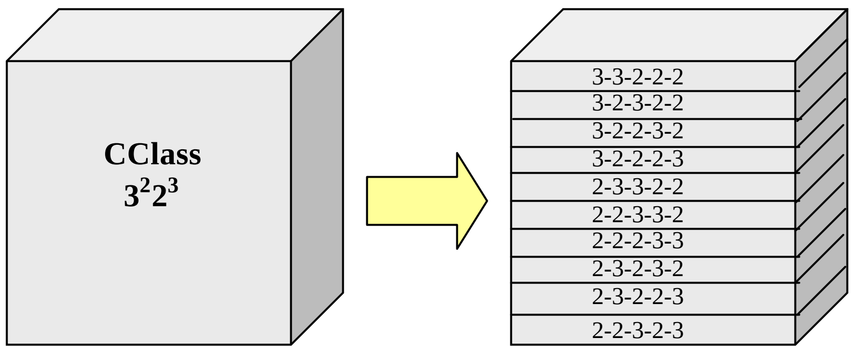

Figure 3.9 Cyclic Form Structure Partition of a Conjugacy Class

The third order partitions are the orbital planes. Orbital planes, as defined in Section 3.5, are created using mutually exclusive groups and are composed of orbits. Orbital planes exclusively and exhaustively partition each cyclic form structure and conjugacy class. The number of orbital planes per cyclic form structure is

$${}_nC_r = \frac{n!}{n!(n-r)!}$$

where *n* is the total number of letters in the groups, and *r* is the total number of mutually exclusive groups. Each cyclic form structure provides for different positioning sequence

combinations for C_i. Therefore, each cyclic form structure has its own unique orbital planes. Figure 3.10 displays the concept of orbital planes within a cyclic form structure.

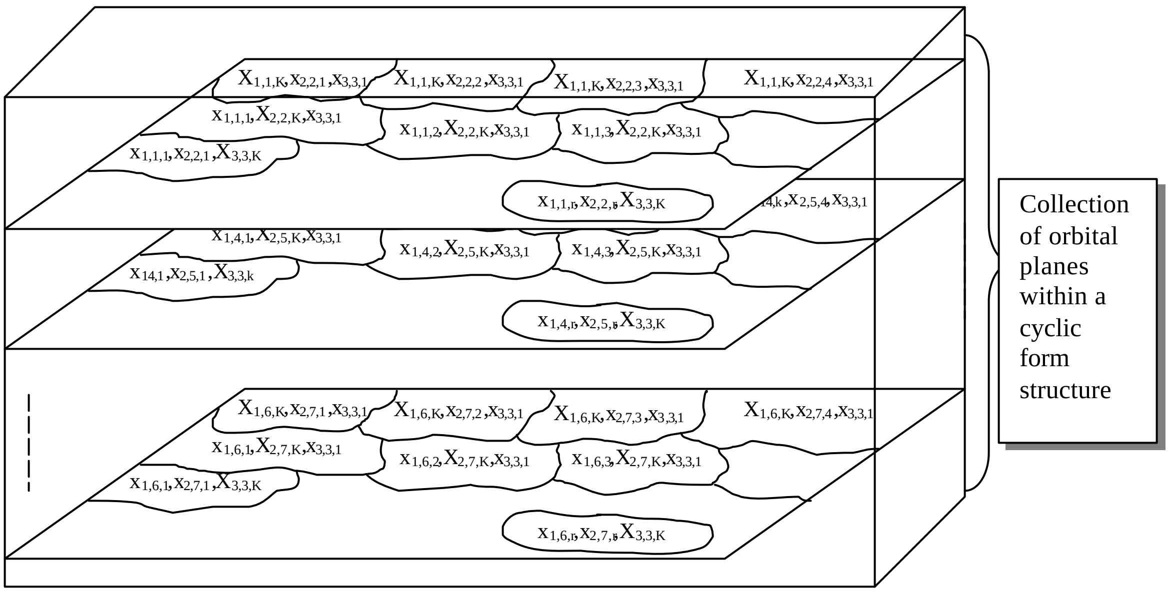

Figure 3.10 Orbital Planes Within a Cyclic Form Structure Partition

The fourth order partitions are the orbits, which exclusively and exhaustively partition orbital planes. The number of orbits within an orbital plane is dependent on the size of the group generating the orbital plane. If a single group generates the orbital plane, then the number of orbits generated by the group G_i is

$$|\text{Orbits}| = \frac{(\prod_{j=1}^{n} | G_j |)}{| G_i |}$$

such that n is the number of mutually exclusive groups and $|G_i|$ is the number of elements in a specific group G_i. G_j are the other groups that make up the orbital plane.

The solution space partition hierarchy provides information on how a TDVRSP space is characterized. The next section demonstrates how the GTTS uses group theory moves to traverse different partitons of the space.

3.7 Traversing S_n with GTTS

This section describes how the TDVRSP solution space is traversed using orbits, orbital planes, cyclic form structures and conjugacy classes. First, some notation and definitions are provided, followed by an algorithm and example.

Let p_i represent the incumbent solution. Index i is the tabu search iteration for each cycle. Let T represent all 2-cycles in S(C) whose letters come from different C_i, i.e., $(m,n) \in T$ implies m and n come from distinct C's. Let G_i = S(C_i). Let λ represent a real-valued function on $S(V \cup C)$. For $Y \subseteq S(V \cup C)$, BEST[Y] denotes a Y-element for which $\lambda(y)$ is optimal in Y.

A general algorithm that traverses the solution space is presented below. Steps 1 through 4 represent a tabu search cycle, and steps 2, 3 and 4 represent the tabu search iterations between cycles.

1. Choose initial incumbent $p_0 \in X(k)$ for the tabu search cycle
2. p_i = BEST[$p_{i-1}^{G_i}$], i = 1..n (declare each orbit tabu). Unless BEST compares an orbit's local optimum to the best-yet solution, $\{\lambda(p_i)\}$ need not be a strictly improving sequence.
3. p_0 = any nontabu BEST[p_n^T];
4. If at any time p_i breaks a non-improving solution threshold, then make a diversification move that changes cycle structure to yield $p_{i+1} \in X(j{\neq}k)$. The search may later revisit $X(k)$.
5. Repeat steps 2 & 3 until break threshold is met.

A generalized example follows. Let p = initial solution. Let C_i = disjoint customer letters $\{C_i : i = 1,2,3,4\}$. Figure 3.11 graphically portrays the following example.

Example

Start tabu search with a p, say the solution is represented as $x_{1,1,1}, x_{2,2,1}, x_{3,3,1}, x_{4,4,1}$.

An iteration is the evaluation of an orbit or transposition neighborhood.
A cycle encompasses one iteration of each G_i and one T. Diversification moves (step 4) may occur within cycles.

Search Cycle 1

Iteration 1 searches neighborhood (orbit) p_0^{G1} and picks p_1 as the new incumbent.
Orbit1 is represented as $X_{1,1,K}, x_{2,2,1}, x_{3,3,1}, x_{4,4,1}$. Assume k = 2 for p_1.

Iteration 2 searches neighborhood p_1^{G2} and selects p_2 as the new incumbent.
Orbit2 is represented as $x_{1,1,2}, X_{2,2,K}, x_{3,3,1}, x_{4,4,1}$. Assume k = 2 for p_2.

Iteration 3 searches neighborhood p_2^{G3} and selects p_3 as the new incumbent.
Orbit3 is represented as $x_{1,1,2}, x_{2,2,2}, X_{3,3,K}, x_{4,4,1}$. Assume k = 2 for p_3.

Iteration 4 searches neighborhood p_3^{G4} and selects p_4 as the new incumbent.
Orbit4 is represented as $x_{1,1,2}, x_{2,2,2}, x_{3,3,2}, X_{4,4,K}$. Assume k = 2 for p_4.

Iteration 5 searches neighborhood p_4^T and selects p_0 as the new incumbent.
Assume the selected swap occurred between C_1 and C_2, where 5 and 6 represents the new positions. $p_0 = x_{1,5,2}, x_{2,6,2}, x_{3,3,2}, x_{4,4,2}$ and is located in a new orbital plane.

Search Cycle 2

Iteration 1 searches neighborhood (orbit) p_0^{G1} and picks p_1 as the new incumbent.
Orbit5 is represented as $X_{1,5,K}, x_{2,6,2}, x_{3,3,2}, x_{4,4,2}$. Assume k = 3 for p_1.

Iteration 2 searches neighborhood p_1^{G2} and selects p_2 as the new incumbent.
Orbit6 is represented as $x_{1,5,3}, X_{2,6,K}, x_{3,3,2}, x_{4,4,2}$. Assume k = 4 for p_2.

Iteration 3 searches neighborhood p_2^{G3} and selects p_3 as the new incumbent.
Orbit7 is represented as $x_{1,5,3}, x_{2,6,4}, X_{3,3,K}, x_{4,4,2}$. Assume k = 3 for p_3.

Iteration 4 searches neighborhood p_3^{G4} and selects p_4 as the new incumbent.
Orbit8 is represented as $x_{1,5,3}, x_{2,6,4}, x_{3,3,3}, X_{4,4,K}$. Assume k = 1 for p_4.

Iteration 5 searches neighborhood p_4^T and selects p_0 as the new incumbent.
Assume the selected swap occurred between C_2 and C_3, where 7 and 8 represents the new positions. $p_0 = x_{1,5,3}, x_{2,7,4}, x_{3,8,2}, x_{4,4,1}$ and is located in a new orbital plane.

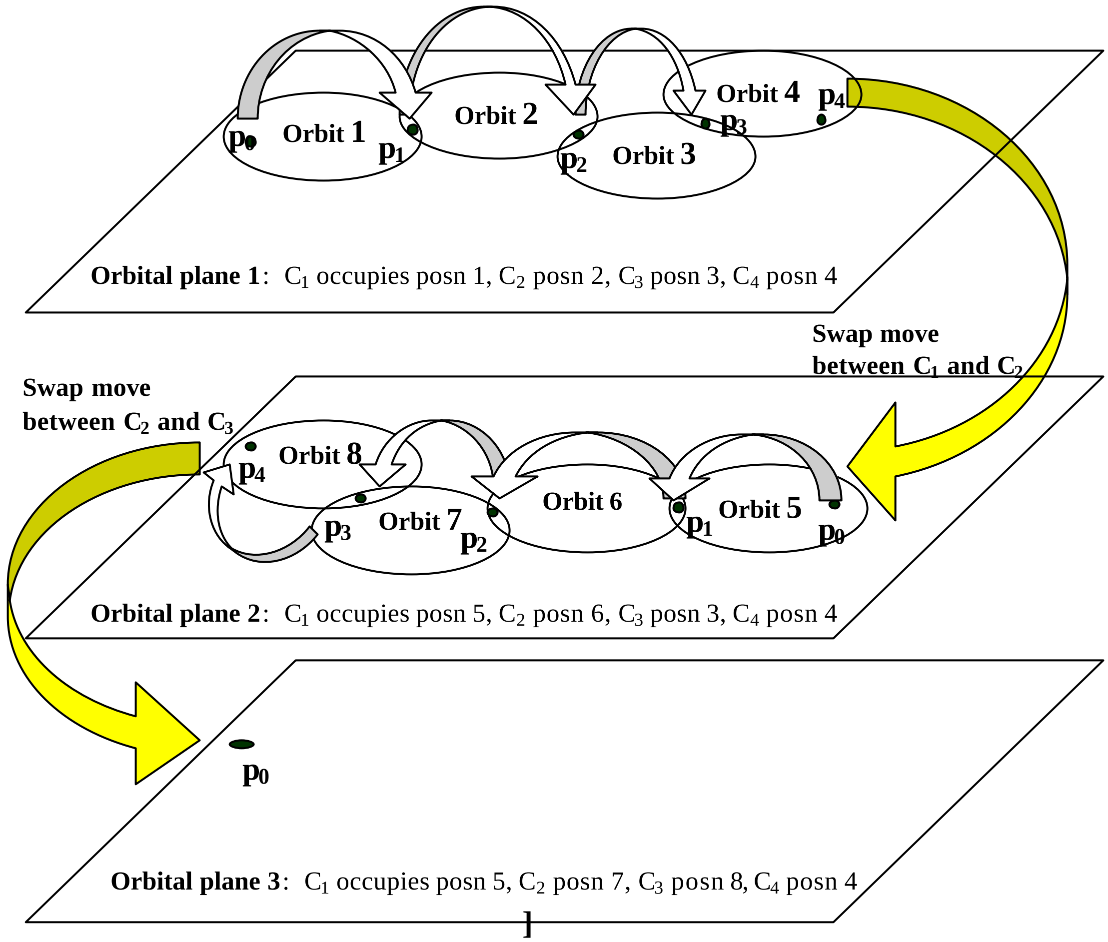

Figure 3.11 GTTS Orbital-plane Traversals Example

The GTTS continues moving through the solution space as presented above until a diversification parameter is triggered. The orbital planes are not kept tabu and the search is free to move between them to improve the solution evaluation. It is possible to move back and forth between two orbital planes as the algorithm searches for better solutions. However, orbits are kept in an indefinite tabu status. Once an orbit is exhaustively evaluated, the search is not permitted to reevaluate the orbit. This prevents cycling and unnecessary computation time.

A non-improving move counter is maintained to trigger diversification procedures. The diversification move comes in two forms, insert moves and extraction moves. Both move types drive the search into a cyclic form structure where the GTTS search cycles resume. The new cyclic form structure may be in the same conjugacy class or different conjugacy class depending on the type of move.

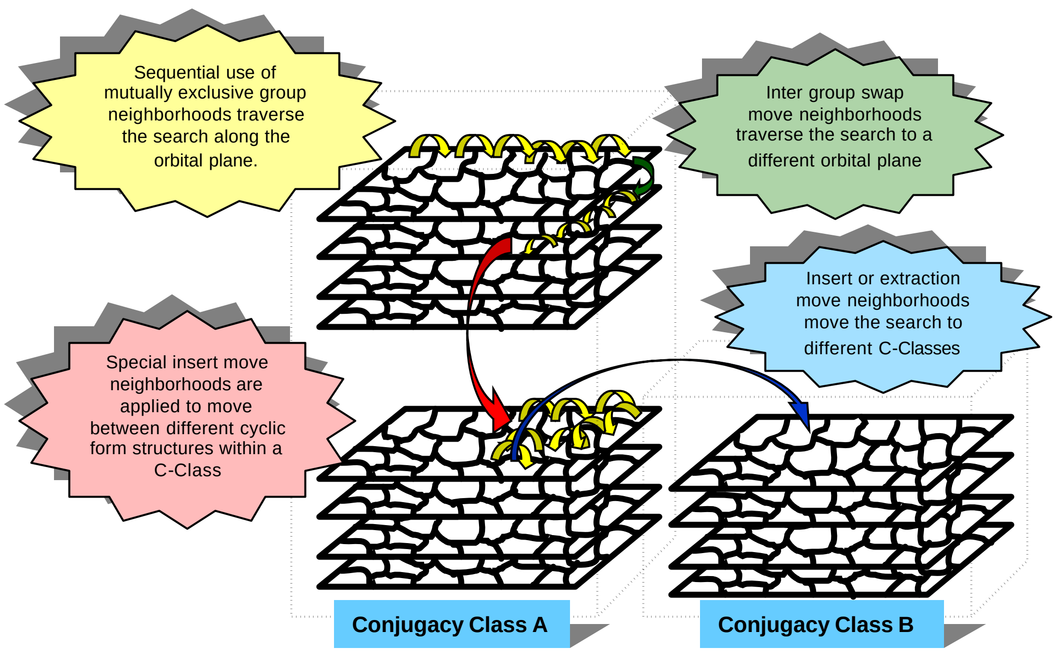

Figure 3.12 Move Effects on S_n Solution Space

Figure 3.12 graphically displays the GTTS as it traverses through S_n solution space. As demonstrated in the previous example, sequential use of mutually exclusive group neighborhoods traverse the search along the orbital plane. The only way the search moves off the orbital plane is by executing an inter-group swap move, insert move, or extraction move. The inter-group swap move neighborhood moves the search to a different orbital plane, but it keeps the search within the same cyclic form structure. There are two types of insert move neighborhoods within the GTTS. One insert move

neighborhood diversifies the search outside of the current cyclic form structure and into a different one, but maintains the same conjugacy class. The other insert move neighborhood moves the search to a different cyclic form structure within a different conjugacy class. The extraction move neighborhood is created to move the search into a different conjugacy class.

3.8 Summary

Group theoretic tabu search has great promise for solving the TDVRSP as it has many advantages over conventional tabu search approaches. First, the solution space can be divided into a hierarchy of partitions. Partitions are searched or avoided based on specific problem characteristics and solution quality. Second, group theory actions direct the search through the space and avoid cycling during the search. Third, a structured move process can be implemented that has the ability to exhaustively search portions or all of the solution space.

IV. A Generalized GTTS Algorithm For The TDVRSP

4.1 Introduction

The tabu search philosophies of adaptive memory and intelligent search integrate effectively with the concepts of group theory. Group theory supports tabu search moves, move neighborhoods, and tabu lists. Group theory also partitions the solution space to better direct and manage the search.

In Section 4.2, a tabu search architecture is described, and in Section 4.3, a generalized GTTS algorithm is presented that solves the TDVRSP.

4.2 Tabu Search Architecture

In this section, a general GTTS architecture, used to solve the TDVRSP, is presented. This architecture, displayed in Figure 4.1, is partitioned into the pre-tabu search phase and the tabu search phase. Listed beneath Figure 4.1 are the major components of each procedure within each phase. These components perform specific functions of the search. The GTTS architecture is based on Harder (2000).

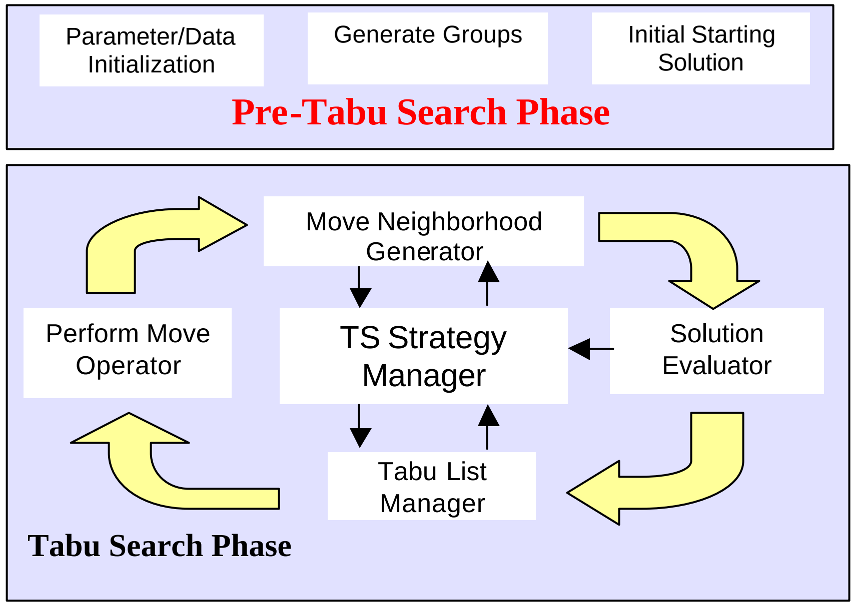

Figure 4.1 Group Theoretic Tabu Search Architecture

1. **Parameter/Data Initialization**
   a. Import data files and preprocess data
   b. Initialize algorithmic parameters
2. **Generate Group Neighborhoods**
3. **Obtain Initial Starting Solution**
4. **Move Neighborhood Generator**
   a. Intra-orbital plane mutually exclusive group action neighborhood
   b. Inter-orbital plane swap move neighborhood
   c. Fill demand insert move neighborhood
   d. Inter-conjugacy class insert move neighborhood
   e. Intra-conjugacy class insert move neighborhood
   f. Inter-conjugacy class extraction move neighborhood
5. **Solution Evaluator**
   a. Vehicle loader/scheduler heuristic
   b. Objective function evaluator
   c. Solution comparison method
6. **Tabu List Manager**
   a. Orbits traversed (optional)
   b. Conjugacy classes traversed (optional)
   c. Move list
7. **Perform Move Operator**
   a. Conjugate
   b. Product
8. **Tabu Search Strategy Manager**
   a. Intensification: focus search within an orbital plane and/or cyclic form structure
   b. Diversification: move to different solution space partition

4.2.1 Pre-Tabu Search Phase

In the pre-tabu search phase, a number of processes initialize the tabu search. First, user inputs dictate the search parameter values defined in Tables 4.1 and 4.2. Table 4.1 displays numerical parameters that prescribe sizes, ranges, and number of process loops. Table 4.2 displays boolean parameters that characterize the TDVRSP instance and what algorithm variant is used. Parameters are discussed in Sections 6.2.1, 6.3.1, and 8.2.

Table 4.1 Tabu Search Parameters (Numerical)

Parameter Name	Definition	Default
periodLength	Actual time period duration to model	None
sizeGroups	Number of letters for the group neighborhoods	5
neighborhoodSizeLimit	Maximum size of each diversification move neighborhood.	500
dataSet	Data set number to run	None
iterations	Number of normal tabu search iterations to run between TS intensification cycles	250
intensificationIterations	Number of intensification tabu search iterations to run between normal TS cycles	50
maxLoops	Number of normal tabu search cycles and intensification tabu search cycles	10
eliteListSize	Number of solutions to keep in the elite list	1
loadDistributionOption	Method used to distribute load to customers	1
worseningMoveTolerance	Number of worsening moves allowed before diversification	3
constantMoveTolerance	Number of constant solution values allowed before diversification	3
intensificationWorsening Move Tolerance	Number of worsening moves allowed before diversification when intensifying	5
intensificationConstant Move Tolerance	Number of constant solution values allowed before diversification when intensifying	5
objFunctionWeights	The weights used to prioritize the different variables in the objective function	{1,1,1,1}
conjugacyClassTabuTenure	Conjugacy class tabu tenure	3
moveTabuTenure	Move tabu tenure	3
superDiversifyRange	Range to check for total cost solution values	200
superDiversifyTolerance	Max number of similar total cost solutions allowed	20
superDiversifyMoves	Number of consecutive diversification moves	6

Table 4.2 Tabu Search Parameters (Boolean)

Parameter Name	Definition	Default
allowConjugateTabuList	Use a conjugacy class tabu list	False
useOrbitTabuList	Use orbit tabu list	True
allowRedundantMoves	Allow moves where the new solution equals the incumbent solution	True
AllowRedundantMoves Intensification	Allow moves where the new solution equals the incumbent solution when intensifying	True
earlyTDDIsHard	Customers have ETDD constraints	False
storageConstraint	Hubs have storage constraints	False
routeLengthConstraint	Vehicles have route length constraints	False
timeWindow Constraint	There exist time windows for customers / hubs / and depots	False
fuelResupplyAvailabe	Customers/hubs have fuel to resupply vehicles	False
allowMOGConstraint	Customers have MOG constraints	False
allowPUMOGConstraint	Depots / hubs have MOG constraints	False

Following parameter initialization, TDVRSP data are imported. The text file formatted data provides customer and vehicle specifications, which are imported and placed into a number of arrays, array lists and vectors. Table 4.3 displays vehicle data types and Table 4.4 displays data for customers, hubs, and depots.

Table 4.3 Vehicle Data Types

Vehicle ID	Speed	# trips /period	Coordinate location
Capacity	Hub service time	Fixed cost	Variable cost
Vehicle type	Available time	Cruising length	Direct Delivery
Depot/hub #	Load time	Unload time	

Table 4.4 Nodal Data Types

Customer ID	Coordinate location	Demand	TDD requirements
Working MOG ground	Working MOG aircraft	Parking MOG aircraft	Early TDD time constraints
Depot or hub	Tier location	Customer priority	Storage capacity
Ground fuel storage	AC fuel storage	Time windows	

The next step in the pre-tabu search phase is the generation of groups. This process partitions the customer service letters C into disjoint subsets C_i , i=1,2,…,n. Different customers are represented in each subset. The process then generates mutually exclusive groups $G_i = S(C_i)$. The generated groups represent all possible permutations of S_n for each subset C_i. Therefore, if $m=|C_i|$, then $|G_i| = m!$. For example, let $C_1 = \{6, 7, 8, 9\}$, then G_1 is generated by group generators (6, 7, 8, 9), (6, 7), which provides 24 permutations within the group action. If $C_2 = \{10, 11, 12, 13, 14\}$, then G_2 is generated by group generators (10, 11, 12, 13, 14), (10, 11) and results in 120 permutations within G_2. Each G_i is calculated and stored within the *move neighborhood generator* for future use as move neighborhoods. Calculating G_i a priori saves computational time during the tabu search phase.

The final step in the pre-tabu search phase is creating and evaluating an initial solution. Assigning prioritized customers to vehicles that can best fill customer demands creates the initial solution. Customers are prioritized by a weighted calculation of demand levels, distance from the nearest depot, and time constraints. Vehicles are ordered based on their capacity and average trip time.

$$CustomerPriorityRating = \frac{custDist}{avgDist} * \sum_{i=1}^{n} [custDemand_i * (periodLength / TDD_i)(n-i)]$$

$$VehCapPerAvgTripTime = vehCap/(2* avgDist/speed + loadTime + unloadTime + servTime)$$

where i = TDD requirement index per customer
n = # of TDD requirements per customer
TDD_i = customer TDD requirement for index i
$custDist$ = distance of customer to nearest depot/hub
$avgDist$ = average distance of all customers to their nearest depot
$custDemand_i$ = customer demand
$periodLength$ = total model time period
$vehCap$ = vehicle capacity

speed = vehicle cruising speed
loadTime = time to load a vehicle
unloadTime = time to unload a vehicle
servTime = time to service a vehicle

The following describes the initial solution heuristic. Vehicles in order of best capacity per trip time fill customer demands by priority. Customers are iteratively assigned to vehicles until all demand is filled or all vehicles are used. The algorithm ensures the appropriate vehicles service customers.

1. Calculate customer priority ratings
2. Sort by customer priority rating (descending)
3. Calculate vehicle capacity per average trip time
4. Sort by vehicle capacity per average trip time (descending)
5. While ($\text{demandshortfall}_{\text{customer}[i]}>0$ and $\text{unfilledCapacity}_{\text{vehicle}[j]}>0$)
   Assign (by priority) letter from customer[i] to trip letter from vehicle[j]
6. End

Once created, the initial solution is evaluated and instantiated as the incumbent solution. The objective function measures the demand filled shortfall, TDD shortfall, fixed costs, variable costs and other penalties. This particular initial solution method is satisfactory for filling customer demand, but it does not concentrate on vehicle scheduling. This naïve approach does not provide good useable solutions. However, they do provide good starting solutions for the GTTS since most of the customer demand is satisfied. GTTS then reorders letters to better satisfy scheduling requirements and fills in demand shortfall as necessary.

4.2.2 Tabu Search Phase

Each iteration of the tabu search cycle passes through five major components; *move neighborhood generator, solution evaluator, tabu search strategy manager, tabu list manager*, and *perform move operator*. An iteration begins by generating and applying a move neighborhood to the incumbent solution and ends when the *perform move operator* transforms the current solution into the new incumbent solution. Each component is detailed in Sections 4.2.2.1 through 4.2.2.5.

4.2.2.1 The Move Neighborhood Generator

The *move neighborhood generator* creates and applies move neighborhoods to the incumbent solution. This process generates a collection of new solutions that best meet *tabu search strategy manager* requirements. One of these becomes the new incumbent solution. For example, if the *tabu search strategy manager* dictates finding improving solutions within the orbital plane, the *move neighborhood generator* generates an orbit. If the *tabu search strategy manager* dictates finding a diversification move that improves filling customer demand by moving to a different conjugacy class or cyclic form structure, then a special insert neighborhood is generated that meets that criteria. The types of move neighborhoods generated are:

1. *Intra-orbital plane mutually exclusive group neighborhood*
2. *Inter-orbital plane swap move neighborhood*
3. *Fill demand insert move neighborhood*
4. *Inter-conjugacy class insert move neighborhood*
5. *Intra-conjugacy class insert move neighborhood*
6. *Inter-conjugacy class extraction move neighborhood*

The *intra-orbital plane mutually exclusive group neighborhood* is a collection of move neighborhoods generated by mutually exclusive groups. The mutually exclusive groups are actually generated in the pre tabu search phase, as described in Section 4.2.1. However, each of the group neighborhoods are stored and executed from the *move neighborhood generator* component. The group neighborhoods are the primary search mechanism imposed by the tabu search strategy during normal search cycles. The manner by which they are utilized is described in Sections 3.5 and 3.7, which supports the exploration of orbital planes.

The *inter-orbital plane swap move neighborhood* is a collection of 2-letter swap moves, of which each letter is from a disjoint set of customer letters, C_i $i = 1,2,\ldots,n$. The purpose of this neighborhood is to diversify the search into a new orbital plane, as prescribed by the *tabu search strategy manager*. The 2-letter swap moves are generated in the pre tabu search phase. Once generated, they are stored and executed in the *move neighborhood generator*.

In order to provide a potentially exhaustive exploration of all orbital planes within a cyclic form structure, there must exist at least one swap move connecting each of the disjoint C_i letter sets, where the minimum number of swap moves is $n-1$. For example, if letter sets C_1, C_2, C_3, and C_4 have four letters each, a minimum of three swap moves is required to connect the sets. Each swap move contains an arbitrary letter from each C_i.

$$\begin{array}{ll}
C_1 = \{w_1,w_2,w_3,w_4\} & \underline{\text{swap moves}} \\
 & (w_i, x_i) \\
C_2 = \{x_1, x_2, x_3, x_4\} & \\
 & (x_i, y_i) \\
C_3 = \{y_1, y_2, y_3, y_4\} & \\
 & (y_i, z_i) \\
C_4 = \{z_1, z_2, z_3, z_4\} &
\end{array}$$

For the *inter-orbital plane swap move neighborhood*, two different size neighborhoods are generated depending on the TDVRSP instance size. For TDVRSP instances where the number of customer letters m is generally less than 200, the number of generated swap moves is

$$|\,swapmoves\,| = \frac{m(m-1)}{2} - \sum_{i=1}^{n} \frac{k_i(k_i - 1)}{2}$$

where $k_i = |C_i|$

This provides a maximum number of swap possibilities between disjoint C_i letter sets. For these swap neighborhoods, each $c \in C_i$ is swapped with another $c \in C_j,\ i \neq j$. Letters are not swapped within C_i because that swap move would not move the search to another orbital plane, and is redundant with a $g \in G_i$. For example, using letter sets C_1, C_2, C_3, and C_4, there exist

|swapmoves|=16*15/2- [(4*3/2)+ (4*3/2)+ (4*3/2)+ (4*3/2)] = 96 swap moves.

$C_1 = \{w_1, w_2, w_3, w_4\}$
$C_2 = \{x_1, x_2, x_3, x_4\}$
$C_3 = \{y_1, y_2, y_3, y_4\}$
$C_4 = \{z_1, z_2, z_3, z_4\}$

Swap move examples:

$(w_1,x_1),(w_1,x_2),\ldots,(w_1,z_4)$ = 12 moves
$(w_2,x_1),(w_2,x_2),\ldots,(w_2,z_4)$ = 12 moves
....
$(x_1,y_1),(x_2,y_2),\ldots,(x_2,z_4)$ = 8 moves
....
$(y_4,z_1),(y_4,z_2),\ldots,(y_4,z_4)$ = 4 moves
total = 96 moves

This method is adequate for generating the *inter-orbital plane swap move neighborhood* for smaller size TDVRSP instances, but it is not adequate for larger scale instances. For instance, if m = 600, n = 120, and k = 5, i = 1,2,…,n, then the number of

swap moves = 178,500, which is not efficient, e.g., 178,500 moves are evaluated to select one move. Therefore, a method that generates *inter-orbital plane swap move neighborhoods* for large TDVRSP problem instances is needed.

For TDVRSP instances where the number of customer letters m generally exceeds 200, the number of swap moves generated is

$$|\,swapmoves\,| = \frac{n(n-1)}{2}$$

where n is the number of C_i letter sets

This is a reasonable number of swap moves for large problems and provides more swap move combinations than the minimum requirement n-1. For this neighborhood, an arbitrary $c \in C_i$ for each C_i performs a letter swap with an arbitrary $c \in C_j$ for each C_j, i *and* $j = 1,2,\ldots,n$ and $i \neq j$. For example, using letter sets C_1, C_2, C_3, and C_4, there exist

|swapmoves|= 4*3/2 = 6 swap moves.

$$C_1 = \{w_1, w_2, w_3, w_4\}$$
$$C_2 = \{x_1, x_2, x_3, x_4\}$$
$$C_3 = \{y_1, y_2, y_3, y_4\}$$
$$C_4 = \{z_1, z_2, z_3, z_4\}$$

One possible set of swap moves is:

$$\{(w_1, x_1), (w_2, y_2), (w_3, z_3), (x_1, y_1), (x_2, z_2), (y_3, z_1)\} = 6 \text{ moves}$$

Although not as extensive as the previous method, this method accommodates an exhaustive search of all orbital planes when combined with group neighborhoods and provides reasonable sized neighborhoods for large problem instances. For example, if $m = 600$, $n = 120$, and $k = 5$ for each i, $i=1,2,\ldots,n$, then the number of swap moves is 7,140 which is much more reasonable than the 178,500 using the other method.

The *fill demand insert move neighborhood* is a collection of moves that help to reduce the incumbent solution's demand shortfall amount. The neighborhood is recreated each time it is called from the *tabu search strategy manager*. The method that builds the *fill demand insert move neighborhood* utilizes the incumbent solution's attributes to create moves that specifically decrease the solution's demand shortfall. The *tabu search strategy manager* specifically calls it when the requirement to diversify into a new cyclic form structure is present in order to fill a solution demand shortfall. The method is partitioned into three phases:

Phase 1. Evaluate the current solution and determine customers with demand shortfall.
Phase 2. Locate vehicles that are not used or have partial loads
Phase 3. Create insert moves, which insert a customer letter found in phase 1 into a vehicle letter cycle found in phase 2.

In phase 1, the incumbent solution is evaluated to calculate the *demandShortfallArray* and *vehicleLoadArray*. If the *demandShortfallArray* [*i*] >0, then customer *i* has a demand shortfall. In phase 2, unused and partially loaded vehicles are found using the following: if *vehicleCapacity*[*j*] – *vehicleLoadArray*[*j*] > 0, then vehicle letter *j* is not fully loaded. Phase 3 is the creation of *inter-cycle insert moves*, which move letters from one cycle to another cycle.

For example, let p = (1,5)(2)(3,6)(4)(7) be the incumbent solution. Let customer *A*, with service letters {5,6,7}, have a demand shortfall. Let vehicles *A* and *B* have respective trip letters {1,2} and {3,4}. In phase 1, p is evaluated which results in identifying customer *A* with a demand shortfall. In phase 2, vehicle trip letters 2 and 4 are identified as empty vehicles. In phase 3, six insert moves are generated. These

moves, when performed on p, result in the following solutions. Group theoretic notation for performing an *inter-cycle insert move* is described later in the *perform move operator*. Generated *fill demand moves* are both *inter-conjugacy class* and *intra-conjugacy class* move types.

moves	$p \oplus move$
insert letter 5 to letter 2	(1)(2,5)(3,6)(4)(7)
insert letter 6 to letter 2	(1,5)(2,6)(3)(4)(7)
insert letter 7 to letter 2	(1,5)(2,7)(3,6)(4)*
insert letter 5 to letter 4	(1)(2)(3,6)(4,5)(7)
insert letter 6 to letter 4	(1,5)(2)(3)(4,6)(7)
insert letter 7 to letter 4	(1,5)(2)(3,6)(4,7)*

* different conjugacy classes

The *inter-conjugacy class insert move neighborhood* consists of *inter-cycle insert moves* that diversify the search by moving into a new conjugacy class. The neighborhood is recreated each time it is called by the *tabu search strategy manager*. The method that builds the *inter-conjugacy class insert move neighborhood* uses the incumbent solution's attributes to create moves that potentially decrease the solutions' demand shortfall or time definite delivery shortfall. It is specifically called during the diversification process when the solution has a time definite delivery shortfall and the strategy calls for moving to a different conjugacy class. The *inter-conjugacy class insert move neighborhood* is generated using two phases:

Phase 1. Collect all the customer letters that occupy unit cycles
Phase 2. Create insert moves, which insert the phase 1 customer letters into appropriate vehicle letter cycles.

Creating the move neighborhood considers the tier structure of the TDVRSP instance as well as customer presence within the cycles. Customer letters are only assigned to vehicles within the same tier, since they are the only vehicles that can deliver to the customers. Also, equivalent customer service letters are not inserted into the same cycle. Finally, customer letters are inserted at the end of the cycle.

For example, let p = (1,7)(2)(3,6)(4)(5)(8) be the incumbent solution. Next, assign customers *A* and *B* with respective service letters {5,6} and {7,8}. Let vehicle *A* and *B* have respective trip letters {1,2} and {3,4}. The insert moves are listed below. Note that customer letter 5 is not inserted in the same cycle as customer letter 6 and customer letter 8 is not inserted in the same cycle as letter 7.

moves	p⊕move
insert letter 5 to letter 1	(1,7,5)(2)(3,6)(4)(8)
insert letter 5 to letter 2	(1,7)(2,5)(3,6)(4)(8)
insert letter 5 to letter 4	(1,7)(2)(3,6)(4,5)(8)
insert letter 8 to letter 2	(1,7)(2,8)(3,6)(4)(5)
insert letter 8 to letter 3	(1,7)(2)(3,6,8)(4)(5)
insert letter 8 to letter 4	(1,7)(2)(3,6)(4,8)(5)

The neighborhood is a collection of *inter-cycle insert moves*, which is defined in the *perform move operator*.

The *intra conjugacy class insert move neighborhood* consists of *inter-cycle insert moves* that traverse the search to a new cyclic form structure but maintains the search within the same conjugacy class. This neighborhood is generated when the *tabu search strategy manager* desires searching within the same conjugacy class. The *intra-conjugacy class insert move neighborhood* is created using the cyclic structure of the incumbent solution.

Generating the move neighborhood considers the tier structure, where customer letters are only assigned to vehicles within the same tier. If a customer letter is already in a cycle, then a customer letter that represents the same customer will not be inserted into that same cycle. Unlike the *inter-conjugacy class insert move neighborhood*, customer letters are inserted at all possible positions within the insert cycle.

In order to maintain the same conjugacy class, a general rule is applied when inserting customer letters to cycles. First, no unit cycle letters are inserted because this will change the conjugacy class. Second, a customer letter from a cycle of size n may only be removed from that and inserted to a cycle of size n-1.

Next, assign customers A and B with respective service letters {5,6} and {7,8}. Let vehicles A and B have respective trip letters {1,2} and {3,4}. For example, let p = (1,7)(2)(3,6,8)(4,5) be the incumbent solution. The insert moves are listed below. Note customer letter 5 from a 2-factor is only inserted to a unit cycle. Customer letter 6 is not inserted in the same cycle as customer letter 5. Customer letter 7 is only inserted to a unit cycle. Customer letter 8 is not inserted in the same cycle as customer letter 7. All permutations maintain the conjugacy class $3^1 2^2 1^1$.

<u>moves</u>	<u>$p \oplus move$</u>
insert letter 5 to letter 2	(1,7)(2,5)(3,6,8)(4)
insert letter 6 to letter 1	(1,6,7)(2)(3,8)(4,5)
insert letter 6 to letter 7	(1,7,6)(2)(3,8)(4,5)
insert letter 7 to letter 2	(1)(2,7)(3,6,8)(4,5)
insert letter 8 to letter 4	(1,7)(2)(3,6)(4,8,5)
insert letter 8 to letter 5	(1,7)(2)(3,6)(4,5,8)

The *inter-conjugacy class extraction move neighborhood* consists of extraction moves that diversify the search by moving into new conjugacy classes. The neighborhood

is generated each time it is called by the *tabu search strategy manager*. The method that builds the *inter-conjugacy class extraction move neighborhood* extracts customer letters from cycles and places them into unit cycles. It is specifically called when excess customer letters reside in cycles and when the *tabu search strategy manager* implements super-diversification measures. A super-diversification measure is further described in section 4.2.2.5.

The method that generates this move neighborhood simply locates each customer letter in a cycle and extracts the letter from that cycle. The customer letter is then placed in a unit cycle. For example, let p = (1,7)(2)(3,6,8)(4,5) be the incumbent solution, customer A and B have respective service letters {5,6} and {7,8} and vehicle A has trip letters {1,2,3,4}. The extraction moves are listed below.

moves	p⊕move
extract letter 5	(1,7)(2)(3,6,8)(4)(5)
extract letter 6	(1,7)(2)(3,8)(4,5)(6)
extract letter 7	(1)(2)(3,6,8)(4,5)(7)
extract letter 8	(1,7)(2)(3,6)(4,5)(8)

In summary, the *move neighborhood generator* creates neighborhoods based on the guidance of the *tabu search strategy manager*. Some of the neighborhoods are generated in the pre-tabu search phase, such as the *intra-orbital plane mutually exclusive groups neighborhood* and *inter-orbital plane swap move neighborhood*. Others are created as the search progresses based on the incumbent solution attributes. Some of the neighborhoods support the intensification process, while others support a diversification process. Once the neighborhoods are generated, they are sent to the *solution evaluator*

process, which is the next topic of discussion. Table 4.5 summarizes the different move neighborhoods.

Table 4.5 Move Neighborhood Summaries

Move Neighborhoods	Move types	Purpose	Move affect	Generated
Intra-orbital plane groups	Groups	Explore orbital planes	Ordering	Pre-Tabu Search. Provides basic TS move neighborhoods
Inter-orbital plane swap moves	2-letter swap moves	Diversify to new orbital plane	Ordering	Pre-Tabu Search. Provides mechanism to explore different orbital planes within same cyclic structure
Fill demand insert moves	1-letter insert moves	Diversify to new conjugacy class	Partitioning	As directed by TS strategy manager to fill customer demands
Inter-conjugacy class insert moves	1-letter insert moves	Diversify to new conjugacy class	Partitioning	As directed by TS strategy manager to reduce TDD shortfall
Intra-conjugacy class insert moves	1-letter insert moves	Diversify w/in a conjugacy class	Partitioning	As directed by TS strategy manager for diversification
Inter-conjugacy class extract moves	1-letter extraction moves	Diversify to new conjugacy class	Partitioning	As directed by TS strategy manager to reduce fixed and variable costs and for super diversification.

4.2.2.2 The Solution Evaluator

The *solution evaluator* determines the objective function value for each *p*-neighbor, denoted q where $q = p \oplus move$ or $q = p^{move}$ $\forall$ $move \in MoveNeighborhood$. The goal of this process is to find the best non-tabu q to replace the incumbent solution p. This process works closely with the *tabu list manager*.

There are four main components of the *solution evaluator*. They are: permutation preprocessor, vehicle loader/scheduler heuristic, objective function evaluator, and solution comparison method. The *solution evaluator* iterates through the four components for all $move \in MoveNeighborhood$.

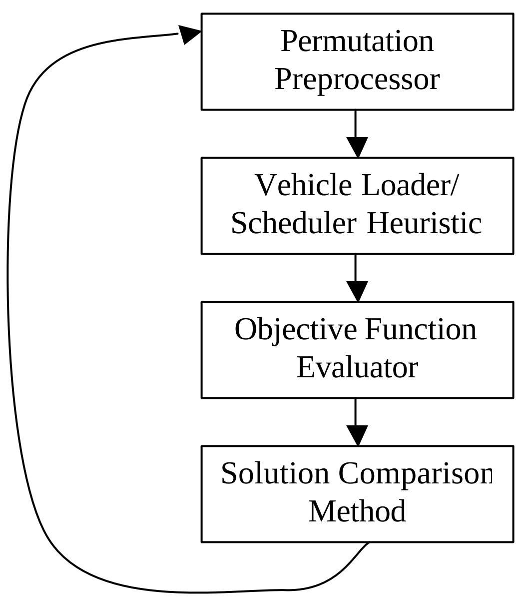

Figure 4.2 Solution Evaluator Component Cycle

The permutation preprocessor converts the incumbent permutation into a neighboring permutation by conducting an operation on the incumbent solution with a *move* from the *MoveNeighborhood*. The type of operation performed is dependent on the

type of *MoveNeighborhood* generated in the previous process. If the neighborhood is a group or swap neighborhood, then conjugation, denoted p^{move}, is performed. If the neighborhood is an insert or extraction move, then the product operation, denoted $p \oplus move$, is performed. Group theoretic specifics on each operation are further discussed in the *perform move operator*. The new permutation q, where $q = p \oplus move$ or $q = p^{move}$, is then sent to the vehicle loader/scheduler heuristic for processing.

The vehicle loader/scheduler heuristic determines the loads each vehicle carries and the vehicle delivery schedules. The heuristic uses the information in permutation q to prescribe vehicle loads and schedules. That information includes each vehicle's routing sequence and customers served within the route. Detail of the heuristic is provided in Section 4.3.

The heuristic utilizes constraint programming techniques to determine the loads and schedules for each vehicle. A vehicle is loaded to its specified capacity unless no more supply exists at the depot/hub or the customers in the tour do not demand a full load. The amount of supply customers receive is prescribed by one of two options. The first option is to provide all the supply a customer demands based on their order within a tour. The second option is to equally split the vehicle load. If a customer does not demand its share, then its share is distributed among the remaining customers. Vehicle schedules are prescribed based on travel times, service times, time constraints, and working MOG constraints. A vehicle schedule begins at its available time and is influenced thereafter by the time constraints along its trip. For example, a vehicle's departure time is adjusted to conform to time constraints that restrict departures during a

specified time window. Also, a vehicle cannot begin unloading at a customer if the working MOG is full.

The vehicle loader/scheduler heuristic's end product consists of two array lists containing customer and vehicle objects that contain solution details. Within each customer object are details of the servicing vehicles' visit. Within each vehicle object are details of the vehicle trip. The array lists are sent to the objective function evaluator to determine the value of the proposed solution.

The objective function evaluator uses information from the vehicle loader/scheduler heuristic to determine the objective function value for each permutation. The objective is to minimize the demand shortfall, time definite delivery (TDD) shortfall, fixed costs, variable costs, and penalty costs. The demand shortfall is the difference between the customer's total demand and the sum total amount of demand delivered to the customer. The TDD shortfall is the weighted difference between the customer's desired delivery time and late delivery time for a set amount of demand. Fixed cost is the total cost for all vehicles used to deliver goods and services. Variable costs are the total costs for vehicles to travel the prescribed routes. Penalty costs include parking MOG violations and hub storage capacity violations. Representative equations used in the *solution evaluator* for each objective function variable are provided below.

d_c = total demand requirement for customer c
$d_{c,t}$ = demand requirement for customer c at time requirement t
dd_c = total demand delivered for customer c
$dd_{c,t}$ = demand delivered for customer c and time requirement t
$cdd_{c,t}$= cumulative demand delivered for customer c and time requirement t
$tdd_{c,t}$ = time demand was delivered for customer c and time requirement t
f_v = fixed cost for vehicle v
a_v = cost per mile for vehicle v

r_v = route lengths for vehicle v
m_c= MOG allowance for customer c
p_c = number of parked vehicles at customer c
s_h = maximum hub storage allowance
cs_h = maximum overage of storage in hub at any given time
n = number of customers
s = number of vehicles
k = number of hubs
m = number of time requirements per customer c
T_t = time for time requirement t
c = customer
t = time requirement
h = hub

$$\textit{demand filled shortfall} = \sum_{c=1}^{n}(d_c - dd_c) \ \ \textit{when } d_c\,^{3}\,dd_c$$

$$\textit{TDD shortfall} = \sum_{j=1}^{2} TDD_j \ \ \textit{where}$$

$$TDD_1 = \sum_{c=1}^{n}\sum_{t=1}^{m}(tdd_{c,t} - T_t) * dd_{c,t}/10 \quad \textit{if } \ (d_{c,t} - cdd_{c,t})\ ^{3}\ dd_c \text{ and } tdd_{c,t}\ ^{3}\ T_t$$

$$TDD_2 = \sum_{c=1}^{n}\sum_{t=1}^{m}(tdd_{c,t} - T_t) * (d_{c,t} - cdd_{c,t})/10 \ \ \textit{if } \mathrm{dd_c} > (d_{c,t} - cdd_{c,t})\ ^{3}\ 0 \text{ and } tdd_{c,t}\ ^{3}\ T_t$$

$$\textit{Fixed cost} = \sum_{v=1}^{s} f_v \ \textit{when } v \textit{ is active}$$

$$\textit{Variable cost} = \sum_{v=1}^{s} a_v r_v$$

$$\textit{MOG penalty} = \sum_{c=1}^{n}(p_c - m_c) \ \ \textit{when } p_c > m_c$$

$$\textit{Storage penalty} = \sum_{h=1}^{k}(cs_h - s_h) \ \ \textit{when } cs_h > s_h$$

Total cost = demand filled shortfall + TDD shortfall + fixed cost + variable cost + MOG penalty + storage penalty

Once all the costs and penalties are computed, they are weighted and placed in an objective function array, denoted $q_{objFunc}$. The weights are parameters assigned to each objective function value. The weighted values are positioned in the array for lexicographic comparison. The order of values in the array are: total cost, demand filled shortfall, TDD shortfall, fixed cost, variable cost, MOG parking penalty, and storage penalty.

The solution comparison method then compares each q objective function and tabu status against the *qBEST* objective function and tabu status as each q_i cycles through the solution evaluator. This process is performed in Harder's (2000) singleThreadedTabuSearch class. The method works with the *tabu list manager* to determine if selecting q is tabu. The permutation q is tabu if Orbit (G_i, q) is tabu, *move* is tabu, or CClass (S_n, q) is tabu. For the following algorithm, the notation $q_{tabuStatus}$ represents each tabu list type. The following algorithm is used to find *qBEST*.

1. $qBEST = q_o$
2. For q_i , $i = 1,2,\ldots, n$ and $n = |MoveNeighborhood|$
   a. If $q_{objFunc} < q_{objFunc}BEST$
      If $q_{tabuStatus} \geq q_{tabuStatus}BEST$
      Then $qBEST = q_i$
   b. Else
      If $q_{tabuStatus} > q_{tabuStatus}BEST$
      Then $qBEST = q_i$

The solution evaluator continues to cycle through the four components for each q generated by the move neighborhood. The permutation *qBest*, determined by the solution comparison method is chosen as the new incumbent solution for the next tabu search iteration.

4.2.2.3 The Tabu List Manager

The *tabu list manager* interacts with the *solution evaluator* to prevent cycling within the tabu search process. Without a tabu list, it is likely in most instances to cycle between permutations and lose algorithmic efficiency. There are two primary functions of the tabu list process; they are to make an object tabu and to check an object's tabu status. For the TDVRSP GTTS, there are three types of tabu lists. They are the orbit, move, and conjugacy class lists. The orbit tabu list is a list of traversed orbits. The move tabu list tracks recent diversification moves and the conjugacy class tabu list maintain recent conjugacy classes searched.

Each tabu list is optional, but at least one list should be utilized to prevent cycling. The orbit tabu list is 100% effective in preventing cycling, but it is computationally more expensive than the other two lists. The move tabu list provided empirical evidence in cycle prevention. Although it is not theoretically 100% effective in preventing cycling, no cycling occurred during testing with the 39-benchmark problems presented in Chapter 5. It is recommended that at least one of these two lists be used. The conjugacy class tabu list is also effective in preventing cycling; however, the list tends to force greater diversification than the other two lists.

Making orbits, moves, and conjugacy classes tabu requires a list for each and a method to represent each object for storage in the list. Table 4.6 displays the tabu list types and objects for each.

Table 4.6 Tabu Lists

Tabu List	List Type	Object Representation	Tabu Tenure
Orbital Tabu List	HashTable	Integer, SymmetricGroup	Indefinite
Move Tabu List	ArrayList	Symmetric Group	User Setting (3)
CClass Tabu List	Array List	Integer	User Setting (3)

The orbital tabu list is a hash table composed of integer and symmetric group objects. The hash table key is the integer object and the hash table object is a symmetric group object. The key is an integer value that represents the orbit and is calculated using the equation below. The hash table object is a symmetric group object representing permutation *p*, which generates Orbit (G_i, p). Dictionary ordering is a necessary condition for this hash code method.

rand = random integer array, size n where $n = |S_n|$
c = customer service letter array
j = customer service letter
$cycle_j$ = disjoint cycle index array for customer service letters for $perm_i$
m = index of cycle position for customer service letter *j* for $perm_i$
$perm_i$ = permutation elements $\in$ Orbit (G_i, p) , $i = 1,2,\ldots,t$. $t = |G_i|$

$$minHashInt = \text{minimum value of } \{ \sum_{j=c_1}^{c_n} rand_j * [(cycle_j + 1) * 10 + m] \,\forall\, perm_i \}$$

$$maxHashInt = \text{maximum value of } \{ \sum_{j=c_1}^{c_n} rand_j * [(cycle_j + 1) * 10 + m] \,\forall\, perm_i \}$$

Hash table integer value = *minHashInt* + *maxHashInt*

The equation generates integer values for each *permutation* $\in$ Orbit (G_i, p) by multiplying a random integer, specifically generated for each customer service letter,

with a unique position value within the cyclic form structure. The *maxHashInt* + *minHashInt* value is used to represent the orbit as the orbit's integer key. A theoretical lower bound collision rate, using the birthday paradox equation, is represented in Figure 4.3. The birthday paradox equation is displayed below.

$$collisionRate = \frac{n(n-1)}{2*|Range|}$$

Where
n = number of compared entities (hash code values)
|*Range*| = cardinality of hash code value range (-2,147,483,647 to 2,147,483,647)

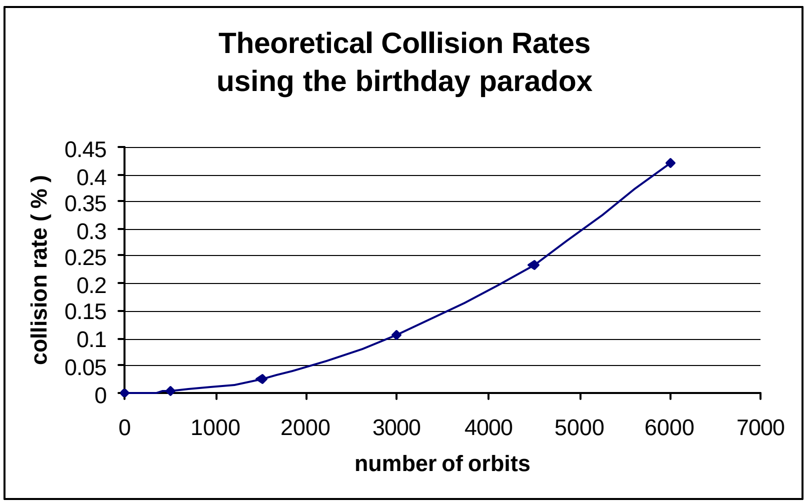

Figure 4.3 Theoretical Collision Rates for Hash Code Values

The move tabu list is simply an array list of symmetric group objects that represent diversification moves. The default tabu tenure is three and is adjustable with user parameters when initiating the program.

The conjugacy class tabu list is an array list that contains integer objects representing conjugacy classes. The default tabu tenure is three and is adjustable with user controls when initiating the program. The conjugacy class list is optional. Although

it was not used to solve the TDVRSP benchmark data set problems, it was used and tested throughout the research as a means to prevent cycling. It becomes more useful when the tabu search strategy is designed to diversify the search throughout many different conjugacy classes. The equation used to create integer objects representing the conjugacy class of permutation p follows:

$$\text{Integer} = \sum_{i=1}^{v} \begin{cases} 10^{2(length_i - 1)} & if \quad length_i > 2 \\ 10^{(length_i - 1)} & if \quad length_i = 2 \end{cases}$$

where $length_i$ is the i^{th} cycle length
v is the max vehicle trip letter

For example, let $p \in$ CClass $4^2 3^4 2^5$ where a cyclic form structure that represents the CClass is (x,x,x,x)(x,x,)(x,x,x)(x,x)(x,x,x),(x,x)(x,x,x,x)(x,x)(x,x,x)(x,x)(x,x,x). The integer value = $(10^{2*3}+10^0+10^{2*2}+10^0+10^{2*2}+10^0+10^{2*3}+10^0+10^{2*2}+10^0+10^{2*2})$

= (1000000 + 1 + 10000 + 1 + 10000 + 1 + 1000000 + 1 + 10000 + 1 + 10000)

= 2040005

4.2.2.4 The Perform Move Operator

After the move neighborhood is generated and evaluated, the best q is selected as the new incumbent solution. The *perform move operator* generates the new incumbent solution using the group theoretic *conjugate* or *product* operation. *Conjugate* operations are used for group and swap move types. The *product* operation is used for the insert and extraction moves.

The *conjugation* operation reorders letters without altering the cyclic form structure of a permutation in S_n. The notation for conjugation is $p^{move} = move^{-1} \oplus p \oplus move$. This represents the conjugate of p by *move.* The result is incumbent solution q.

For example, let p = (0,9,28,29,27)(1,10,26) and *move* = (9,10). The result q is a reordering of letters 9 and 10.

$$p^{move} = (0,9,28,29,27)(1,10,26)^{(9,10)} = (0,10,28,29,27)(1,9,26) = q$$

The *product* operation provides the ability to insert or extract letters from one cycle to another. Inserting or extracting letters between disjoint cycles is called the *inter-cycle insert move.* This operation results in creating a new cyclic form structure. The *inter-cycle insert move's* formulation, in generalized group theoretic terms, is presented below. For this move, we want to insert subpath $[a,\ldots,b]$ from cycle p to cycle q, such that the subpath is placed after letter x. The formulation uses the tour (gpq) where γ are the disjoint cycles not affected by the insertion, p is the cycle losing subpath $[a,\ldots,b]$, and q is the gaining cycle. The formulation is presented below.

$$\text{new solution} = gpq \oplus (a, (b)p, (x)q)$$

To illustrate this move, let p = (0, 9, 28, 29, 27) and q = (1, 10, 26). The subpath is [9] and x is letter 10.

$$\gamma(0, 9, 28, 29, 27)(1, 10, 26) \oplus (9, 28, 26) = \gamma(0, 28, 29, 27)(1, 10, 9, 26)$$

As you can clearly see, the subpath [9] was moved from cycle p to q. As a result, the cycle structure of the solution has changed.

4.2.2.5 The Tabu Search Strategy Manager

The *tabu search strategy manager* determines which solution space partition to search and whether to intensify or diversify the search. Decisions are logically based on collected search data and pre-defined search parameters. Intensification occurs in orbital planes and/or cyclic form structures that generate good solutions. Diversification occurs when the search process needs to explore different solution space partitions.

The *tabu search strategy manager* utilizes listeners and counters within the tabu search process to collect search data. Model parameters, inputted in the pre-tabu search phase, are used as thresholds within the *tabu search strategy manager*. For each tabu search iteration, collected data are compared with parameters that determine whether to continue with the normal search, intensify, or diversify. The listeners collect data for the worsening solutions and new incumbent solutions events.

The worsening solution listener collects the *worsening solution counter*. Data collected for the new incumbent solutions are the *super diversification counter*, *constant solution counter*, and *solution evaluation array*s. Data used in the *tabu search strategy manager* that are not collected by listeners are the *tier counter*, *group action counter*, *tabu search iteration counter*, and *intensification counter*.

The *worsening solution counter* monitors the number of worsening solutions that are selected as new incumbent solutions. A worsening solution is defined as the lexicographic comparison $p_{objFunc}new > p_{objFun}old$ where $p_{objFunc}new$ is the objective function array for the new incumbent solution and $p_{objFun}old$ is the objective function array for the old incumbent solution. Once the *worsening solution counter* ≥ *worseningMoveTolerance*, a diversification move is performed and the counter is reset to

zero. The counter also resets to zero whenever the new *best solution listener* is instantiated.

The *constant solution counter* counts the number of constant value solutions that are selected as new incumbent solutions. A constant value solution is defined as total cost comparison $p_{totalcost}new = p_{totalcost}old$ where $p_{totalcost}new$ is the total cost value of the new incumbent solution and $p_{totalcost}old$ is the total cost of the old incumbent solution. Once the *constant solution counter* $\geq$ *constantMoveTolerance*, a diversification move is performed and the counter is reset to zero. The counter also resets to zero whenever $p_{totalcost}new \neq p_{totalcost}old$.

The *super diversification counter* counts the number of solutions that have similar total cost values within a tolerance. The default tolerance is set to ±0.01%. The method compares each new incumbent solution's total cost to a block of recent solutions' total costs. The recent solutions are stored in the *solution evaluation arrays*. The *superDiversifyRange* and *superDiversifyTolerance* are prescribed as parameters. When the *super diversification counter* $\geq$ *superDiversifyTolerance*, a series of diversification moves are performed. The number of diversification moves is prescribed in the parameters as *superDiversifyMoves*.

The *tier counter, group counter*, *tabu search iteration counter*, and *intensification counter* are also used to direct the tabu search. The *tier counter* provides information to the *move neighborhood generator* in order to utilize the appropriate letters when creating the move neighborhood. The *group counter* sends information to the *move neighborhood generator* and *tabu list manager* in order to prescribe the appropriate neighborhood and in order to register and check appropriate orbits in the tabu list. The *tabu search iteration*

counter counts the number of tabu search iterations. When *tabu search iteration counter*^{3} *iterations*, the search begins an intensification process. The *intensification counter* tracks the number of intensification iterations. When *intensification counter* ^{3} *intensificationIterations*, the search converts back to the normal tabu search process.

Diversification occurs when the search traverses from one solution space partition to another. Diversification traversals occur when the search departs an orbital plane, cyclic form structure, or conjugacy class. Diversification occurs in two forms in the GTTS: normal and super. Normal diversification is a new search direction that attempts to improve solution values, but it does so in a different solution space partition. Super diversification is a new search direction used to depart local search areas. Empirical evidence indicates that super diversification moves tend to be un-improving.

Upon selecting normal diversification, the *tabu search strategy manager* determines which diversification move neighborhood would best direct the search effort. The manager uses the current incumbent solution attributes for making that selection. Those characteristics include: solution demand shortfalls, TDD shortfalls, unnecessary vehicle trips, and unnecessary customer services. The following logic is used to determine which move neighborhood is selected for diversifying the search.

```
If  solution demand shortfall > 0
        Use the fill demand move neighborhood
Else If  unnecessary vehicle trips > 1 or unnecessary customer services > 2
        Use the inter conjugacy class extraction move neighborhood
Else If  TDD shortfall > 0
        Use the inter-conjugacy class extraction move neighborhood
        And alternate with
        the inter-conjugacy class insert move neighborhood
Else
        Use the inter-orbital plane swap move neighborhood
```

When the demand filled shortfall is > 0, the *fill demand move neighborhood* inserts customers in need of supply to vehicles with available capacity. This helps reduce the solution demand shortfall. When unnecessary vehicle trips and customer services fill the solution, the *inter- conjugacy class extraction move neighborhood* extracts customers that do not need service from vehicles providing them service, resulting in the elimination of unnecessary travel. When the TDD shortfall is > 0, *the inter-conjugacy class extraction move neighborhood* removes customers receiving services beyond their TDD time. Alternating with the extraction move neighborhood is *the inter-conjugacy class insert move neighborhood*, that inserts customers to vehicles to better meet the TDD requirements. When none of the above conditions are met and diversification is warranted, the *inter-orbital plane swap move neighborhood* is used. This neighborhood diversifies the search and keeps it within the same cyclic form structure, thus only reordering the letters and not changing the partition.

Super diversification is used to depart local search areas. The super diversification method is a series of consecutive *inter-conjugacy class extraction move neighborhoods*. The number of consecutive moves is prescribed by *superDiversifyMoves*. The default value is six consecutive moves.

Intensification of a search is the concentrated exploration of orbital planes within a cyclic form. The move neighborhoods used are the *intra-orbital plane mutually exclusive groups neighborhood* and *inter-orbital plane swap move neighborhood*. These move neighborhoods only reorder letters and do not change the cyclic form structure. Search intensification begins by selecting an incumbent solution from the elite solution list. The elite solution list is a list of the *x* best solutions found, where *x* is specified by

the *eliteListSize* parameter. The search primarily uses the *intra-orbital plane mutually exclusive groups neighborhood* to explore the elite solution's orbital plane. Once the *intensification worsening move counter* ≥ *intensificationWorseningMoveTolerance* or *intensification constant move counter* ≥ *intensificationConstantMoveTolerance*, the search conducts a swap move from the *inter-orbital plane swap move neighborhood*. The swap move traverses the search to a different orbital plane where the search continues. Once the *intensification iteration counter* reaches its threshold, the search ends the intensification phase and commences the normal tabu search process.

4.2.2.6 Summary

The tabu search process is iterative and cycles through each of the preceding processes. The *move neighborhood generator* creates move neighborhoods based on the *tabu search strategy manager* direction. The *solution evaluator* evaluates objective functions for each *p*-neighbor, checks the tabu status with the *tabu list manager*, and selects the best *p*-neighbor as the new incumbent solution. The incumbent solution is created by the *perform move operator*. Once the new incumbent move is generated, the tabu search begins the cycle again. The *tabu search strategy manager* collects statistics during the search in order to direct intensification and diversification efforts. Upon reaching the tabu search iteration threshold, the tabu search process ends and results are collected.

4.3 The GTTS TDVRSP Algorithm

Section 4.3 provides pseudocode for the generalized GTTS TDVRSP algorithm. The algorithm is presented as three sub-algorithms: GTTS TDVRSP algorithm preprocessor, GTTS TDVRSP algorithm, and the vehicle loader/scheduler evaluation heuristic. The GTTS TDVRSP algorithm preprocessor is a precursor to the GTTS TDVRSP algorithm. Once completed, it is not repeated. The GTTS TDVRSP algorithm conducts the tabu search process. Within the algorithm, the vehicle loader/scheduler evaluation heuristic is called to assign cargo amounts to the vehicles, schedule the vehicles, and evaluate the solution. The three algorithms are presented below.

GTTS TDVRSP Algorithm Preprocessor

1. Set GTTS algorithm parameters
2. Import data from text files
3. Develop Group Actions
    a. Partition customers service letters by tiers
    b. Partition customer service letters in defined group sizes
    c. Minimize the number of customer service letters that represent the same customer in each mutually exclusive set of letters.
    d. Generate groups for each mutually exclusive set of letters
    e. Store groups in the move neighborhood generator
4. Create Initial Solution
    a. Order vehicles by capacity per average trip time
    b. Order customers by demand, time requirements, and distance
    c. For each vehicle trip letter, assign a customer service letter by the orders specified in a. and b.
    d. Repeat step c. until all vehicle trip letters are assigned or all customer demands are met.

GTTS TDVRSP Algorithm (for each iteration)

1. Select MoveNeighborhood
   a. If *super diversification counter* $\geq$ *superDiverfyTolerance* and *diversifyingCounter* $\leq$ *superDiversifyMoves*
      i. Select *inter- conjugacy class extraction move neighborhood*
   b. Else If *worsening move counter* $\leq$ *worseningMoveTolerance* or *constant move counter* $\leq$ *constantMoveTolerance*
      i. Select next sequential group neighborhood, or
      ii. If not strictly intensifying in an orbital plane
         Select swap move neighborhood at end of group sequence
   c. Else If intensificationOn = true
      Select swap move neighborhood
   d. Else
      i. If solution demand shortfall > 0
         Select *fill demand move neighborhood*
      ii. Else If unnecessary vehicle trips > 1 or customer services > 2
         Select *inter- conjugacy class extraction move neighborhood*
      iii. Else If TDD shortfall > 0
         1. Select *inter- conjugacy class extraction move neighborhood*, or
         2. Select *inter- conjugacy class insert move neighborhood*
      iv. Else
         Select *inter- orbital plane swap move neighborhood*

2. Evaluate $q = p\text{Å}move$ *or* $q = p^{move}$ $\forall$ $move \in$ *MoveNeighborhood*. Call detailed vehicle loader/scheduler evaluation heuristic.
3. Select best q based on objective function values and tabu status
4. Register evaluated orbit, conjugacy class, and/or move in tabu list
5. Perform $p\text{Å}move$ *or* p^{move} and generate a new incumbent solution, p
6. Determine tabu search type, normal or intensification
   a. If *tabu search counter* = *iterations*,
      i. Begin intensification search process
      ii. Set p = an elite list permutation
      iii. *IntensificationOn* = true
   b. Else If *intensification tabu search counter* = *intensificationIterations*,
      i. Begin normal tabu search process
      ii. Set *intensificationOn* = false.
7. Repeat steps 1-6 with p for *maxLoops*
8. Print Output
9. End GTTS

Vehicle Loader/Scheduler Evaluation Heuristic

1. For each tier, determine the vehicle delivery schedules and loads for *q*
    a. Get supply object availability times at depot/hub locations,
    b. For each supply object
        i. Assign an available vehicle to deliver the supply. Vehicles are assigned by availability time and in capacityPerTripTime order.
        ii. If vehicle route length and refueling means < max route length or vehicle type is not service compatible to customer
            1. Do not load or schedule vehicle and return to step b.i.
        iii. Vehicle load = max{vehicle capacity, available supply, tour demand}. Tour demand applies when tour demand< capacity.
        iv. Vehicle load is distributed to customers via option 1 or 2.
            1. Option 1: fill customer's demand by order within the tour.
            2. Option 2: split load among the customers with demand.
        v. Schedule vehicle departure and arrival times
            1. If working MOG constraint is on
                a. Assign start load time and end load time at hub/depot based on MOG constraint.
            2. If time window or ETDD constraints are on
                a. Assign hub/depot/customer departure times based on time windows.
                b. Assign hub/depot/customer arrival time based on time windows/ETDD and travel times.
            3. Else If do not consider TW or ETDD to assign times.
            4. If working MOG constraint is on:
                a. Assign start unload and end unload times at customer based on working MOG constraint.
                b. If MOG constraints delay unload times
                    i. Change the vehicle's departure time.
            5. Repeat steps 2,3 and 4 for each customer in the tour.
        vi. Vehicle is assigned an availability time based on its arrival back to the hub/depot and service time. It waits in an arrival queue until a supply object is ready for onload.
    c. If there exists supply in supply object
        i. Repeat step b with a new vehicle.
    d. If no supply exists in supply object and vehicle has a partial load
        i. Vehicle waits until more supply is available for onload and repeat steps b.ii-b.v
    e. Repeat steps b, c, and d until all supply is exhausted or all vehicle trips are exhausted or all customer demand is satisfied.
2. Calculate the objective function
    a. Determine customer demand shortfall
    b. Determine customer TDD shortfall
    c. Determine fixed and variable costs

d. Determine penalties
e. Assign objective function values to evaluation array

4.4 Summary

Chapter 4 provides a detailed explanation of the GTTS TDVRSP algorithm. The algorithm builds on Harder's (2000) tabu search architecture and incorporates group theoretic moves and move neighborhoods. The two phases of the GTTS TDVRSP are the pre-tabu search phase and tabu search phase. The pre-tabu search phase imports and preprocesses data and initializes algorithmic parameters. The pre-tabu search phase also generates two move neighborhood types and creates an initial starting solution. The tabu search phase is comprised of five components. One cycle of the five components is a tabu search iteration. The five components are the *move neighborhood generator, solution evaluator, tabu list manager, perform move operator,* and *tabu search strategy manager.*

V. Construction of Benchmark Problems

5.1 Introduction

Since benchmark problems did not exist for the TDVRSP prior to this research effort, it was necessary to construct a set of problems that adequately tests the robustness of the GTTS TDVRSP algorithm. A valid set of benchmark problems requires an experimental design that effectively varies the data to account for different TDVRSP instances. Section 5.2 presents the benchmark problem design process for the TDVRSP problems.

There are three traits used to test the TDVRSP algorithm robustness: problem size, problem density, and problem type. Problem size relates to the number of variables in the problem. The number of variables depends on the number of vehicle trip and customer service letters. As these variables increase, the size of the problem increases and the problem becomes more computationally difficult. Problem density relates to the difficulty of satisfying customer demand and time definite delivery requirements. Problem instances come from the theater distribution hierarchy presented in Section 1.4.2. The problems are composed of three TDVRSP types: the Air Force multiple trips multiple services (MTMS) without hub, the Joint MTMS without hub, and the Joint MTMS with hub and other defining constraints.

5.2 Benchmark Problem Design Hierarchy

Figure 5.1 is a hierarchical portrayal of the benchmark design. The first level of the hierarchy is the problem size (small, medium and large). The second and third levels

detail the problem density with delivery restriction density (low, medium, and high) and demand to capacity ratio (low, medium and high). The fourth level details other special problem considerations.

The TDVRSP algorithm is not specifically designed to solve problems with highly variable unrealistic features or characteristics; rather, it is specifically designed to solve typical problems that occur in practice. Hence, the medium ranges portray conditions where supply is consistent with demand, and time requirements are realizable but moderately difficult to achieve. Figure 5.1 characterizes 39 benchmark problems: 12 problems each for the small and medium problem sizes, and 15 problem instances for the large problem size.

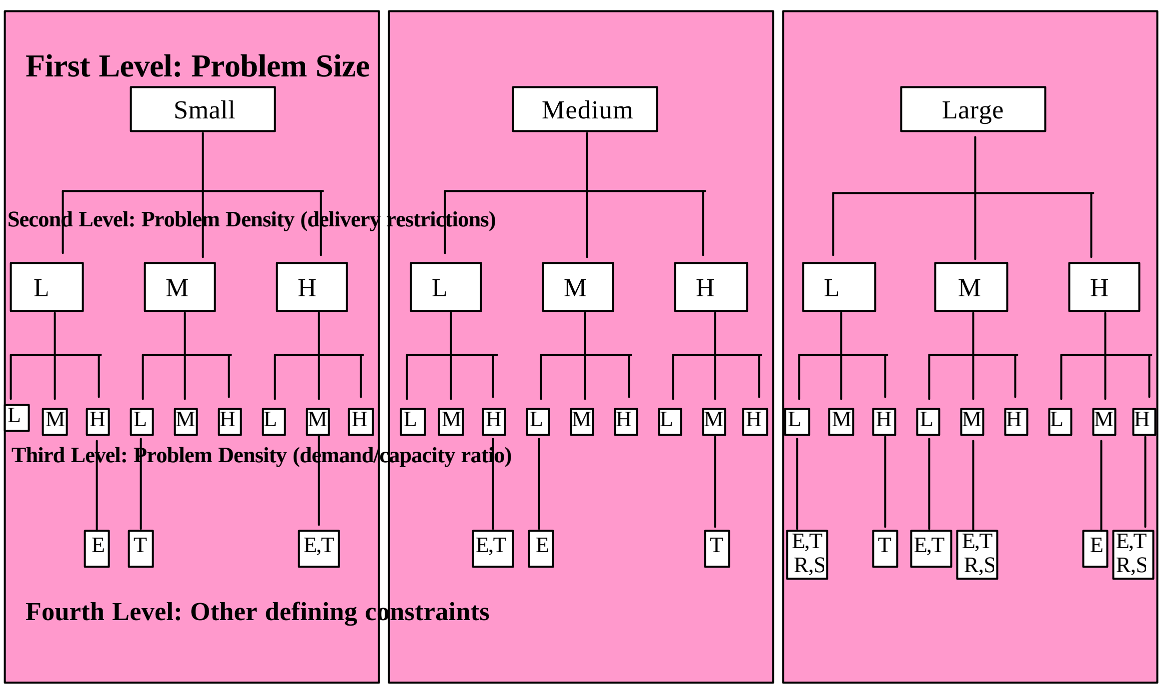

Figure 5.1 TDVRSP Benchmark Problem Hierarchy

5.3 Problem Size, First Level

In the hierarchy's first level, each problem size has nine low-medium-high combinations of the problem density plus additional instances of other defining constraints. Three TDVRSP instances are used to represent small, medium, and large size problems. The small problem is a multiple trip multiple service without hub instance that uses strictly aircraft. The medium problem is a multiple trip multiple service without hub instance that uses air and ground vehicles. The large problem is a multiple trip multiple service with hub instance that uses air and ground vehicles. Table 5.1 provides problem size distributions of customers and vehicle trips.

Table 5.1 Problem Size Distribution of Customers and Vehicle Trips

Problem Size	Number of customers	Total Customer Demand (tons)	Expected Vehicle Trips
Small	8	2000	Low
Medium	8	3000	Low
Large	31	9000	High

A low (high) expected number of vehicle trips ranges from 1 to 5 (6 to 10) and the number of trips depends on total customer demand and the total time period. If there are 20 vehicles with a total capacity of 400, the expected number of trips per vehicle to fill demand of 1000 is 2.5. Demand and time period must be coordinated to allow adequate travel time for the required number of vehicles. The number of vehicle trip and customer visit variables for each problem size is displayed in Table 5.2. The approximate number of variables for each problem size is 170, 350, and 1350 for the small, medium, and large problem sizes, respectively.

Table 5.2 Problem Size Distribution of Variables

Size	# Cust	Demand	# Veh (Cap)	E(trips/veh)	E(# trips)	Max(#visits)	total variables
Small	8	2000	17 (640)	3	70	100	170
Med	8	3000	41 (960)	3	150	200	350
Large	31	9000	90 (1500)	6	600	750	1350

5.4 Problem Density (Delivery Restrictions, Second Level)

Delivery is restricted primarily by working MOG, the number of vehicles that can simultaneously service a customer, and time definite delivery (TDD) requirements, i.e., the times that services must be complete at a customer location. Delivery restrictions have many causes such as materiel handling equipment, limited workforce, or limited docking space. For the purpose of this design, MOG 1 is a high restriction, MOG 2 is a medium restriction, and MOG 3 is a low restriction.

Any service exceeding a TDD requirement at a customer location is late. TDD restrictions are measured by the TDD time to period length ratio, i.e., if the time period length is 72 hours and the TDD time is 36 hours, then the ratio is ½. High restrictions have ratios between ¼ and ½, medium restrictions are between ½ and ¾, and low restrictions are between ¾ and 1.

A combination of MOG and TDD time is used to characterize the customers' delivery restrictions. To make the scenario a low, medium, or high delivery restriction, customers are assigned a MOG and TDD time. In order to avoid assigning all customers the same MOG and TDD characteristics in a single scenario, a distribution plan was used to weigh the number of customers with low, medium and high delivery restrictions. Therefore, Table 5.3 is used to distribute the number of low, medium, and high restrictions for the MOG and TDD to the customer base.

Table 5.3 MOG and TDD Density Distribution Scheme

MOG constraint	%MOG(1,2,3)	TDD constraint	%TDD(1/4-1/2,1/2-3/4, ¾-1)
Low	(25,25,50)	Low	(10, 40, 50)
Medium	(25,50,25)	Medium	(25, 50, 25)
High	(50,25,25)	High	(50, 25, 25)

Tables 5.4, 5.5, and 5.6 are used to calculate the customer MOG and TDD characteristics for each delivery restriction scenario. Once a customer service is characterized in one of the nine groupings, it is assigned a random TDD ratio within the defined parameters and a MOG.

Table 5.4 Low Delivery Restriction Distributions

MOG\TDD	¼ - ½ (10%)	½ - ¾ (40%)	¾ - 1 (50%)
MOG 1 (25%)	.025	.10	.125
MOG 2 (25%)	.025	.10	.125
MOG 3 (50%)	.05	.20	.25

Table 5.5 Medium Delivery Restriction Distributions

MOG\TDD	¼ - ½ (25%)	½ - ¾ (50%)	¾ - 1 (25%)
MOG 1 (25%)	.0625	.125	.0625
MOG 2 (50%)	.125	.25	.125
MOG 3 (25%)	.0625	.125	.0625

Table 5.6 High Delivery Restriction Distributions

MOG\TDD	¼ - ½ (50%)	½ - ¾ (25%)	¾ - 1 (25%)
MOG 1 (50%)	.25	.125	.125
MOG 2 (25%)	.125	.0625	.0625
MOG 3 (25%)	.125	.0625	.0625

Table 5.7 labels the cells as low, medium low, medium, medium high, and high restrictions. Customer services assigned to these cells are grouped into these restriction categories.

Table 5.7 Combined MOG and TDD Density Distribution Scheme

MOG\TDD	¼ - ½	½ - ¾	¾ - 1
MOG 1	High	Medium High	Medium
MOG 2	Medium High	Medium	Med Low
MOG 3	Medium	Med Low	Low

The distributions of low, medium low, medium, medium high, and high customer delivery restrictions are displayed as multinomial distributions in Figure 5.2, Figure 5.3, and Figure 5.4. Notice how the restriction distributions change over the three charts.

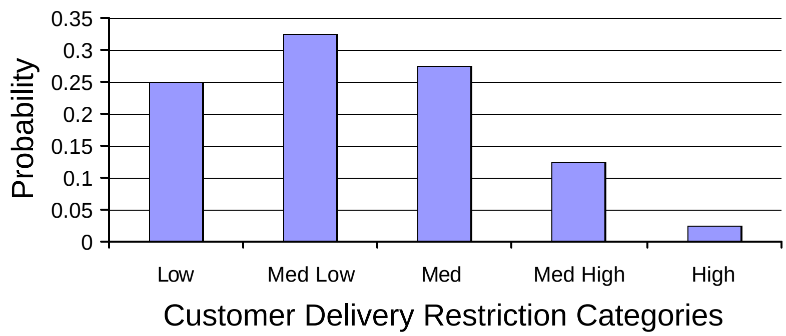

Figure 5.2 Low Delivery Restriction Distribution

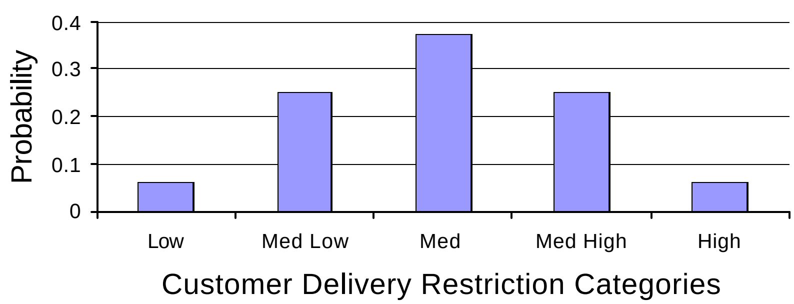

Figure 5.3 Medium Delivery Restriction Distribution

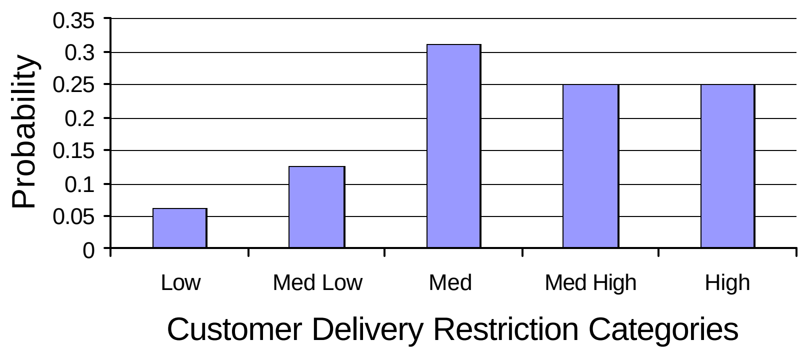

Figure 5.4 High Delivery Restriction Distribution

The delivery restrictions are methodically spread among the customer services. If the TDVRSP has 24 customer service requirements, then the low, medium and high restriction matrices are multiplied by 24 and rounded to the nearest integer. This categorizes each customer service requirement. Once each customer service requirement is allocated into a category, they are assigned their MOG and TDD ratio value. The TDD ratio value is determined randomly between the TDD ratio parameters.

5.5 Problem Density (Demand to Capacity Ratio, Third Level)

The third level in the benchmark problem design hierarchy is the problem density characterized by the total customer demand (in tons) to total vehicle capacity ratio. Total customer demand is the sum of all customer demands over the time period. Total vehicle capacity is defined as the sum of the capacities for all vehicle trips. For example, if Vehicle A has a capacity of 20 tons and has the potential to make three trips, then the sum of the vehicle capacity for vehicle A is 60 tons. The demand to capacity ratios, low, medium, or high, are defined in Table 5.8.

Table 5.8 Demand to Capacity Ratio Parameters

Demand to Capacity ratio	Total demand as a % of total capacity
Low	80% - 90%
Medium	95% - 100%
High	110% - 120%

For example, if the total vehicle capacity is 1000 tons, then a low demand to capacity ratio implies a demand between 800 and 900 tons.

The second and third level implies a 3 by 3 factorial design, providing nine problem densities for each problem size. Problem densities range from low-low through high-high, for a total of nine combinations which should provide problem density robustness for the TDVRSP.

5.6 Other Defining Constraints (Fourth Level)

The "other defining constraints" allow inclusion of constraints not defined in the theater distribution hierarchy such as the early TDD constraint, multiple no delivery time window constraint, route length constraint with refueling option, and hub/depot storage constraints.

The hard early TDD constraint increases problem density by restricting vehicles from servicing customers prior to a specified delivery time. The hard multiple no delivery time window (MTW) constraint increases problem density by restricting vehicles from arriving or departing a customer/hub/depot for a given time period. The route length (RL) constraint limits the vehicle travel distance between refuelings. While refueling may occur at delivery locations, the fuel supply is constrained. The soft hub/depot storage constraint penalizes storage overflow at hub/depot locations by means

of a penalty term in the objective function. The fourth level provides other defining constraint combinations with instances from the first three levels. Other defining constraint instances are:

E:	ETDD instance
T:	MTW instance
E, T:	ETDD and MTW instance
E, T, R, S:	ETDD, MTW, RL, and Storage instance

The instances are distributed among the 27 low-medium-high combinations defined by the first three levels. There is an E, T, and ET combination for each defined problem size such that each combination is dispersed among the low-medium-high problem density combinations. The E, T, R, S combination is only distributed within the large problem size because the storage constraints are only applicable to the MTMS w/hub instance.

5.7 TDVRSP Problems

The problems, presented in Table 5.9, are numbered according to the problem design hierarchy in Figure 5.1. The low-med-high combinations are indexed 1 to 27 and the other defining constraint combinations are indexed 28 to 39. Air Force (AF) problems and Joint (J) problems are specified.

Table 5.9 Problem Instances

Problem	Instance - size	Del. restrictions	Dem/cap ratio	Other constraints
1	AF-MTMS - small	Low	Low	None
2	AF-MTMS - small	Low	Med	None
3	AF-MTMS - small	Low	High	None
4	AF-MTMS - small	Med	Low	None
5	AF-MTMS - small	Med	Med	None
6	AF-MTMS - small	Med	High	None
7	AF-MTMS - small	High	Low	None
8	AF-MTMS - small	High	Med	None
9	AF-MTMS - small	High	High	None
10	J-MTMS - med	Low	Low	None
11	J-MTMS - med	Low	Med	None
12	J-MTMS - med	Low	High	None
13	J-MTMS - med	Med	Low	None
14	J-MTMS - med	Med	Med	None
15	J-MTMS - med	Med	High	None
16	J-MTMS - med	High	Low	None
17	J-MTMS - med	High	Med	None
18	J-MTMS - med	High	High	None
19	J-MTMS w/hub - lg	Low	Low	None
20	J-MTMS w/hub - lg	Low	Med	None
21	J-MTMS w/hub - lg	Low	High	None
22	J-MTMS w/hub - lg	Med	Low	None
23	J-MTMS w/hub - lg	Med	Med	None
24	J-MTMS w/hub - lg	Med	High	None
25	J-MTMS w/hub - lg	High	Low	None
26	J-MTMS w/hub - lg	High	Med	None
27	J-MTMS w/hub - lg	High	High	None
28	AF-MTMS - small	Low	High	E
29	J-MTMS - med	Med	Low	E
30	J-MTMS w/hub - lg	High	Med	E
31	AF-MTMS - small	Med	Low	T
32	J-MTMS - med	High	Med	T
33	J-MTMS w/hub - lg	Low	High	T
34	AF-MTMS - small	High	Med	E, T
35	J-MTMS - med	Low	High	E, T
36	J-MTMS w/hub - lg	Med	Low	E, T
37	J-MTMS w/hub - lg	Low	Low	E, T, R, S
38	J-MTMS w/hub - lg	Med	Med	E, T, R, S
39	J-MTMS w/hub - lg	High	High	E, T, R, S

5.8 Summary

This chapter describes the benchmark problems used to test TDVRSP algorithm robustness. They are detailed in Appendix D.

The first three design levels have three ranges: low, medium and high. The medium range represents moderate size, supply/demand ratio and delivery difficulty. The ranges above and below the medium level are designed to vary the tightness and looseness of the moderate conditions, in order to test algorithmic robustness.

There are numerous other data elements not explicitly described in this chapter. They include many of the vehicle characteristics such as speed, service time, and rest time. Although not explicitly described, they are implicitly included via the expected number of vehicle trips. For this model, the vehicle types are developed to closely resemble military vehicle specifications. The number of vehicles by type is held constant for each problem instance. Another primary data type is the customer locations, which are located within the theater doctrinal layout. Corps, division, and brigade customers are each located in their assigned areas, and areas of operations are assigned according to doctrine.

VI. The Multiple Trips Multiple Services TDVRSP Instance

6.1 MTMS Introduction

The multiple trips, multiple services TDVRSP instance (MTMS), discussed in Section 1.4.2, is applicable to Air Force theater distribution logistics problems and joint command level theater distribution problems. In this chapter, the general MTMS is defined and two specific MTMS types are presented.

6.1.1 General Concept

The MTMS differs from the typical GVRP by the addition of multiple vehicle trips and multiple customer services. A multiple trip is defined as one or more vehicles that depart a depot/hub, service one or more customers, return to the same depot/hub, and repeat the process during a given time period. A multiple service is defined as a customer that is serviced by more than one vehicle during a given time period. The MTMS includes many of the typical GVRP dimensions and constraints such as single or multiple depots, single or multiple homogeneous or multiple nonhomogeneous vehicles, time window constraints, and route length constraints. In addition, the MTMS includes direct delivery vehicles, working MOG and parking MOG constraints. The primary MTMS objectives are to minimize the amount of unmet demand, late delivery, vehicle fixed costs, and vehicle variable costs.

A vehicle tour is the set of trips assigned to the same vehicle. In group theoretic terms, a symmetric group cycle with a vehicle trip letter, v_i, and at least one customer service letter, c_j, constitutes a vehicle trip $(v_i,c_j,\ldots,c_k)$. A vehicle tour is one or more

symmetric group cycles $(v_1,c_j,\ldots,c_k)$ $(v_2,c_a,\ldots,c_c)$ … $(v_i,c_d,\ldots,c_f)$, *where* $v_1,v_2,\ldots,v_i \in V_z$ are the set of vehicle trip letters assigned to vehicle *z*.

A customer service occurs when a vehicle services a customer. Multiple customer services must occur on separate vehicle trips during a specified time period. Customer services are represented as letters within a symmetric group, where $c_j \in C_n$ are the ordered customer service letters assigned to customer *n*.

Figure 6.1 illustrates a MTMS scenario. Letters {1,2,3}, {4,5}, and {6,7,8,9} belong to vehicles A, B, and C, respectively. Letters {10,11}, {12,13,14}, and {15} belong to customers A, B, and C, respectively. The permutation (1,10)(2,11,13)(4,12)(6,15) represents the scenario where vehicle A makes two trips, vehicles B and C make one trip each, customer A receives two services, customer B receives two services, and customer C receives one service.

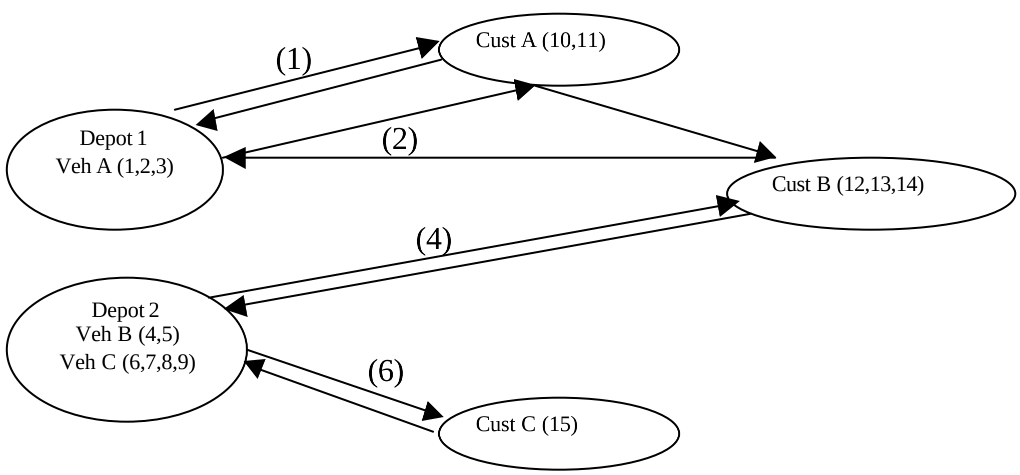

Figure 6.1 MTMS Example

Vehicles are assigned to depots that supply them cargo for delivery. Each vehicle trip starts and ends at the assigned depot. The MTMS allows single or multiple depots Figure 6.1 has one vehicle assigned to depot 1 and two vehicles assigned to depot 2.

MTMS cases allow single vehicle or multiple vehicles and classify vehicle sets as homogeneous or nonhomogeneous. Homogeneous vehicles share common characteristics such as capacity, speed, and vehicle type.

There are three types of time window constraints in an MTMS. They are early time definite delivery (ETDD), time definite delivery (TDD), and multiple time windows for non-departure and non-arrival times (MTW). The ETDD stringently defines customer service starts but does not constrain vehicle arrival or departure times. The TDD is a soft constraint that defines when a customer service should be complete. TDD constraints not achieved are penalized. TDD does not constrain when a customer is serviced, or when vehicle arrival and departure times occur. MTWs are hard constraints that restrict vehicle arrival and departure at a depot/customer. However, they do not constrain when vehicles are on-loaded or off-loaded. The time window constraints are portrayed in Figure 6.2.

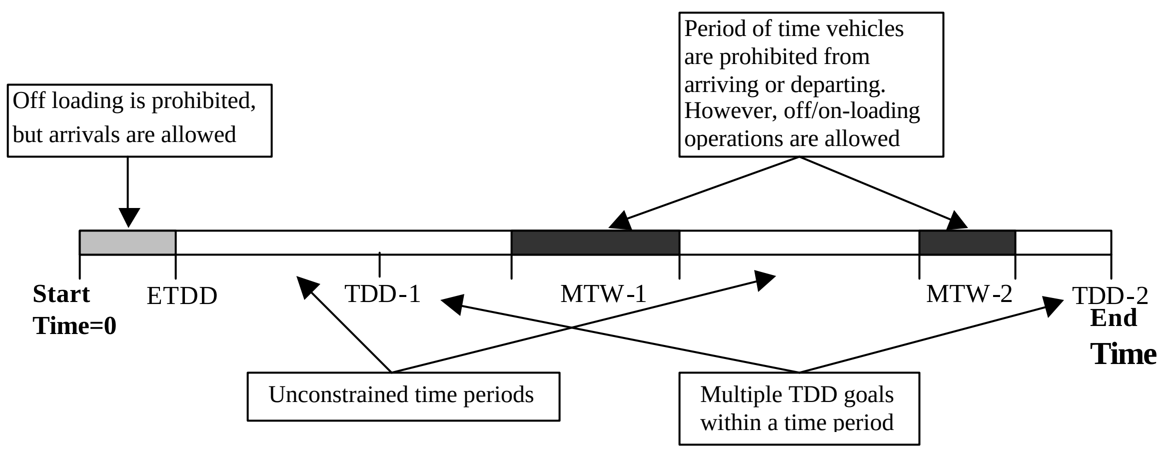

Figure 6.2 MTMS TDVRSP Time Constraints

For the MTMS, route length constraints limit the distance a vehicle may travel before refueling. While refueling may occur at customer locations, it is more common to have vehicles refueled at depots/hubs.

Direct delivery vehicles originate outside the theater and service a customer within the theater. In the MTMS, direct delivery vehicles have a defined arrival time and are fully loaded. Some direct delivery vehicles leave the theater soon after delivering goods, while others are assigned to a depot and continue their trips using the depot as a supply source and originating point. The direct delivery vehicle customers are determined within the tabu search process.

The MOG constraint is used as both a working MOG constraint and parking MOG constraint. Working MOG constrains the number of vehicles that can simultaneously service a customer. Working MOG is working MOGA for aircraft and working MOGG for ground vehicles, to account for their different asset requirements. Parking MOG limits the number of aircraft parked at a customer location. The parking MOG constraint counts vehicles actively waiting their turn to load or unload. The working MOG constraint is hard and the parking MOG constraint is soft. The working MOG is managed as a first-in first-out (FIFO) queue.

The MTMS primary objectives are to minimize unmet customer demand, late deliveries, vehicle fixed costs, and vehicle variable costs. The amount of customer demand not delivered is the demand shortfall. Late deliveries are called TDD shortfall. Late delivery times are weighted by the amount of demand delivered late. Each vehicle incurs a fixed cost when included in the solution. Vehicle variable costs are associated with vehicle route travel. Section 4.2.2.2 presents the mathematical equations used to represent these costs.

6.1.2 Vehicle Characteristics

Table 6.1 presents the explicit vehicle characteristics associated with each unique vehicle for the MTMS. All times are measured in hours.

Table 6.1 MTMS Vehicle Characteristics

Vehicle ID	Average Speed (mph)	Number of trips	Assigned Depot Coordinates	Capacity (Tons)
Service time	Load time	Unload time	Available time	Fixed cost
Variable cost	Cruising length	Depot #	Direct delivery	Vehicle type
Miles/gallon				

Vehicle available time stipulates when a vehicle is available to begin on-loading operations. It usually follows a vehicle service. The service time accounts for refueling operations, crew rest, and other activities between trips. Load time is the time required to load at the depot and unload time is the time required to unload at a customer location. The vehicle schedule is primarily determined by available time, service time, load time, unload time, and speed. As presented in Figure 6.3, a vehicle trip begins with loading at the available time. Following loading, the vehicle departs. Distance/speed determines the travel time between the depot and customers. Unloading follows arrival at each customer location. Trip completion occurs upon return to the depot where the vehicle is serviced. Service completion makes the vehicle available for the next trip. Other factors, discussed in Section 6.1.3, that affect vehicle timelines include time windows, working MOGs, and ETDDs.

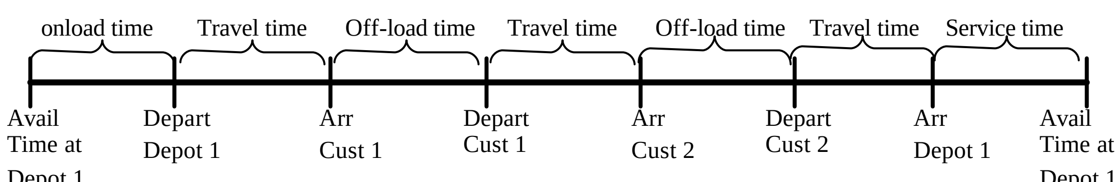

Figure 6.3 Example Vehicle Trip Timeline

Fixed cost is the cost to use a vehicle in the model. It represents the cost to have that vehicle in theater. The costs could be vehicle lease costs, transportation costs, or host nation support costs. The fixed cost is used only one time per vehicle tour. Variable cost is the cost per mile of a vehicle tour. Variable costs include the cost of such things as fuel, wear and tear, maintenance, and labor.

Cruising length is the distance a vehicle can travel on a full tank of fuel. Cruising length is used for determining if and when a vehicle requires refueling. Some customers may have refueling capabilities, while others may not. The cruising length constrains whether or not a vehicle can service a customer. If a vehicle is assigned to service a customer, but the distance to the customer exceeds cruising length, then the vehicle receives no load and does not make the trip.

Depot number is the depot to which the vehicle is assigned. The vehicle departs and returns to the same depot.

Vehicle type specifies whether the vehicle is an air or ground vehicle. It is important to differentiate between vehicle types because customers accommodate vehicle types differently. Some customers have airfields to accommodate aircraft, while others do not.

Direct delivery vehicles are those that begin their trips outside the theater of operations. They service a customer within the theater without stopping at a depot beforehand.

6.1.3 Customer Characteristics

There are a number of explicit customer characteristics within the MTMS instance. The characteristics shown in Table 6.2 are defined for each distinct customer ID.

Table 6.2 MTMS Customer Characteristics

Customer ID	Coordinates	Demand	Time windows	ETDD
Working MOG	Parking MOG	Customer Priority	Fuel storage	TDD

Customer ID is the identification number for each distinct customer. Customer ID numbers range from *0 to m*. The coordinate location is the coordinates of the customer's location. Coordinates are used to determine distances between the depot and the customer locations and between customer locations. Demand is the amount of service the customer requires. Demand is used in two forms; overall demand and demand by time periods.

TDD are the time definite delivery requirements, which specify the time requirement for customer services. There may be multiple TDD requirements for each customer. ETDD is the early time definite delivery requirements. This is a specific no earlier than service time for customers. The multiple time windows are the time periods for which no vehicles can arrive or depart customer locations.

Customer priority defines the importance of servicing a customer relative to the other customers. Priorities range from 0.0 to 1.0, where 0.0 is the lowest priority and 1.0 is the highest priority.

Fuel storage is the amount of fuel on hand to supply vehicles in need of fuel. Vehicles requiring fuel for travel to the next customer/depot draw fuel from the customer's fuel storage.

6.1.4 Assumptions

There are a number of assumptions for the MTMS instance. Most of these assumptions were made to maintain a generalized GTTS TDVRSP algorithm and to permit the creation of generalized TDVRSP benchmark data sets. Too many detailed requirements would cause the algorithm and benchmark data sets to become too specialized for the purpose of this research. However, most assumptions can be eliminated with no or little additional code work. Assumptions that require extensive code work or changes to the algorithm are noted with an *.

1. There is unlimited supply at time 0 at all depot locations.
2. All airfields accommodate all aircraft types
3. Each aircraft type uses the same loading and unloading materiel handling equipment, personnel, and docking space for assessing working MOG constraints.
4. Each ground vehicle type uses the same loading and unloading materiel handling equipment, personnel, and docking space for assessing working MOG constraints.
5. Vehicle schedules do not consider crew rest scheduling*.
6. Unload/load times are static for each vehicle type.
7. Variable costs do not depend on-load amount.
8. Direct delivery vehicles carry a full load.
9. Distance matrix is based on straight-line distances between points.
10. Parking MOGs are soft constraints*.
11. Working MOG rates are constant for each customer and depot*.
12. Vehicles depart from and return to the same depot*.

6.2 Air Force MTMS Instance

There are twelve MTMS instances in the benchmark data set that specifically model only Air Force cargo aircraft. The instances are multiple trip multiple service, single depot, multiple nonhomogeneous vehicle TDVRSP instances with direct delivery vehicles, working and parking MOG constraints, and time window constraints. In this section, benchmark data set 34 and its solution are presented as an example of the Air Force MTMS instance. Figure 6.4 presents the APOD and customer locations relative to each other. The arrow pointing from coordinate (0,0) is the reference point for the direct delivery vehicles. The APOD is approximately 400-500 miles from each customer location. The customers are Corps and Division SSAs that can accommodate Air Force cargo aircraft. Since the distance between the depot and the customers are far less than aircraft cruising distances, route length constraints and refueling requirements are not included in this instance.

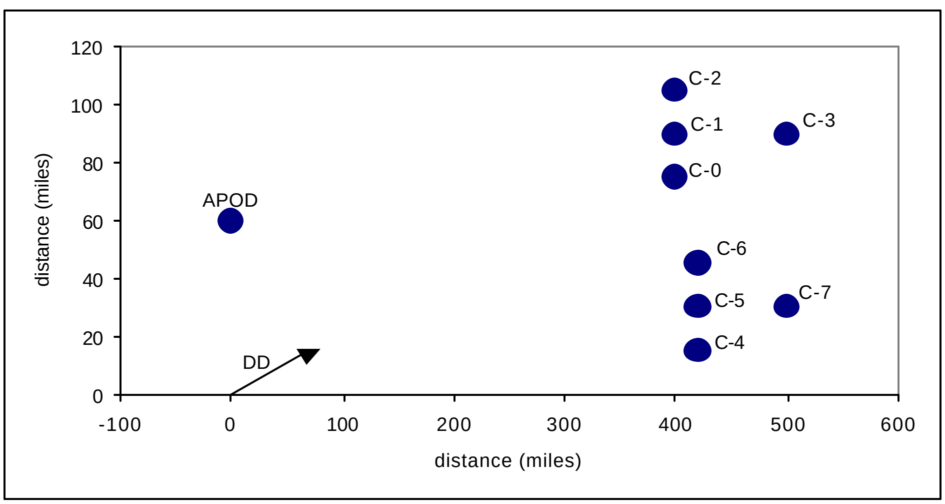

Figure 6.4 APOD and Customer Geographical Distribution

6.2.1 Data and parameter settings

Tables 6.3, 6.4, and 6.5 display customer data used in the model. There are a total of eight customers with varying characteristics. Four customers demand 380 tons and four customers demand 120 tons. The working and parking MOGs vary in size from 1–3 for the working MOGA and 2 – 6 for the parking MOGA. The APOD is restricted to a working MOGA = 4. Customers 0, 2, and 5 have ETDD constraints > 0, while customers 3, 6, and 7 have time windows that restrict arrival and departure times. Table 6.5 presents multiple TDD requirements for each customer. In Table 6.5, the demand is displayed as a cumulative demand, such that those cumulative demand levels are required for each specified TDD requirement. Actual delivery levels less than those specified for the required time are weighted and penalized.

Table 6.3 Customer Data

Cust ID	locationCoor		Demand	wMOGA	pMOGA	ETDD	TDD	Priority
0	400	75	380	1	2	12	46	1
1	400	90	380	1	2	0	48	1
2	400	105	380	2	4	6	42	1
3	500	90	380	3	6	0	48	1
4	420	15	120	1	2	0	48	1
5	420	30	120	2	4	18	46	1
6	420	45	120	3	6	0	44	1
7	500	30	120	1	1	0	48	1

Table 6.4 Customer No-Delivery Time Windows

Customer ID	AC-time windows	
3	48	54
6	18	24
7	24	36

Table 6.5 Customer TDD Requirements

cust ID	cummulative demand	ETDD	TDD
0	174	12	22
0	274	12	35
0	380	12	46
1	85	0	14
1	181	0	22
1	275	0	34
1	380	0	48
2	71	6	18
2	182	6	23
2	264	6	32
2	380	6	42
3	82	0	14
3	175	0	21
3	263	0	33
3	380	0	48
4	28	0	16
4	52	0	22
4	98	0	34
4	120	0	48
5	80	18	30
5	120	18	46
6	30	0	18
6	92	0	36
6	120	0	44
7	25	0	14
7	55	0	24
7	120	0	48

The vehicle data utilized in the Air Force MTMS instance is displayed in Table 6.6. Each vehicle has its own identification number and characteristics. For this instance, there are two aircraft types. The aircraft closely resemble the C-17 and C-130 cargo aircraft. Vehicles 0-5 are the C-17 representatives and vehicles 6-16 are the C-130 representatives. Vehicles 0 and 1 are direct delivery vehicles that originate outside the theater of operations and their available times are > 0. Those same vehicles later become theater assets and are represented as vehicles 2 and 4. The capacity is in tons, speed is in

mph, and service, load, and unload times are in hours. The available time is in hours relative to the model start time. The cruising length is in miles and is not constraining for this problem. The fixed costs and variable costs are hypothetical and are used simply as a means to minimize the use of aircraft and distances traveled.

Table 6.6 Air Force MTMS TDVRSP Vehicle Data

Veh	# Trips	Cap	Speed	Serv Time	Location Coordinates		Fixed Costs	Var Cost	Avail Time	CL	DD	Load Time	Unload Time
0	1	85	470	2	0	0	0	0.05	10	9000	1	4	2
1	1	85	470	2	0	0	0	0.05	20	9000	1	4	2
2	3	85	470	2	0	60	10	0.05	20	9000	0	4	2
3	4	85	470	2	0	60	10	0.05	0	9000	0	4	2
4	2	85	470	2	0	60	10	0.05	30	9000	0	4	2
5	5	85	470	2	0	60	10	0.05	0	9000	0	4	2
6	5	12	330	1	0	60	2	0.02	0	3000	0	2	1
7	5	12	330	1	0	60	2	0.02	0	3000	0	2	1
8	5	12	330	1	0	60	2	0.02	0	3000	0	2	1
9	5	12	330	1	0	60	2	0.02	0	3000	0	2	1
10	5	12	330	1	0	60	2	0.02	0	3000	0	2	1
11	5	12	330	1	0	60	2	0.02	0	3000	0	2	1
12	5	12	330	1	0	60	2	0.02	0	3000	0	2	1
13	5	12	330	1	0	60	2	0.02	0	3000	0	2	1
14	5	12	330	1	0	60	2	0.02	0	3000	0	2	1
15	5	12	330	1	0	60	2	0.02	0	3000	0	2	1
16	4	12	330	1	0	60	2	0.02	12	3000	0	2	1

Tables 6.7 and 6.8 provide the vehicle trip letters and customer service letters used to format the TDVRSP in S_n. There are 69 vehicle trip letters and 120 customer service letters for a total of 189 letters.

Table 6.7 Vehicle Trip Letter Assignments

Vehicle ID	Vehicle Trip Letters
0	0
1	1
2	2,3,4
3	5,6,7,8
4	9,10
5	11,12,13,14,15
6	16,17,18,19,20
7	21,22,23,24,25
8	26,27,28,29,30
9	31,32,33,34,35
10	36,37,38,39,40
11	41,42,43,44,45
12	46,47,48,49,50
13	51,52,53,54,55
14	56,57,58,59,60
15	61,62,63,64,65
16	66,67,68,69

Table 6.8 Customer Service Letters

Customer ID	Customer Letters
0	70,…,89
1	90,…,109
2	110,…,129
3	130,…,149
4	150,…,159
5	160,…,169
6	170,…,179
7	180,…,189

The GTTS parameters used to solve this particular AF MTMS instance are displayed in Table 6.9 and justified in Chapter 8. For this instance, there were a total of 10,000 tabu search iterations. The load distribution option was set at 1. Option 1 is when vehicle ordering determines the amount of service for each customer. The number of worsening moves and constant move tolerances were set at 3 and 5 for normal and intensification tabu search iterations. The super diversification parameters are set to use 6 diversification moves when 20 total cost solution values are within ±0.01% of each

other within a block of 200 solutions. The *allow redundant moves* parameter is turned on, which allows the search to select an identical permutation within the evaluated orbit. The route length was turned off for this instance because aircraft cruising distances are far greater than the theater distances. The storage constraint is turned off because there are no hubs with constraining storage facilities in this instance. An elite list of size 1 was used as the incumbent solution for the intensification iterations. The move tabu tenure is 3, while the orbit tabu tenure is indefinite.

Table 6.9 AF MTMS Parameters and Settings

Parameter	Parameter setting
periodLength	48
sizeGroups	5
neighborhoodSizeLimit	1000
dataSet	34
iterations	150
intensificationIterations	50
maxLoops	50
loadDistributionOption	1
moveTabuTenure	3
worseningMoveTolerance	3
constantMoveTolerance	5
intensificationWorseningMoveTolerance	3
intensificationConstantMoveTolerance	5
eliteListSize	1
superDiversifyRange	200
superDiversifyTolerance	20
superDiversifyMoves	6
objFunctionWeights	{1,1,1, .05}
allowConjugate Tabu List	FALSE
allowRedundantMoves	TRUE
allowRedundantMovesIntensification	TRUE
earlyTDDIsHard	TRUE
storageConstraint	FALSE
routeLengthConstraint	FALSE
fuelResupplyAvailable	FALSE
timeWindowConstraint	TRUE
allowMOGConstraint	TRUE
allowPUMOGConstraint	TRUE

6.2.2 Results

The Air Force MTMS results are summarized in Tables 6.10 and 6.11. Detailed vehicle loads, schedules, and customer deliveries are displayed in Appendix A Tables A.1 and A.2. The solution, in terms of S_n, is displayed below. The conjugacy class of the solution is $2^{61}3^{9}$. Most of the vehicle trips had single customer deliveries and some had multiple customers.

(0 182)(1 133 78)(2 154)(3 102)(4 75)(5 131)(6 103)(7 118)(8 83)
(9 132)(10 134 93)(11 90)(12 111)(13 71)(14 174)(15 164 94)(16149)
(17 113)(18 72)(19 163)(20 128)(21 155)(22 114)(23 80)(24 169 95)
(25 122)(26 117)(27 151)(28 74)(29 105)(30 124)(31 159)(32143)
(33 119)(34 161)(35 125)(36 170)(37 86)(38 121)(39 109)(40 115)
(41 70 96)(42 126)(43 160)(44 110 98)(45 180)(46 73)(47 97)(48 92)
(49 123)(50 181)(51 147)(52 112)(53 79)(54 100)(55 120)(56173)
(57 144)(58 167 99)(59 129)(60 183)(61 172)(62 76)(63 165)
(64 162 175)(65 88)(66 82)(67 127)(68 106)(69 116 77)

For the best-found solution, there was a demand shortfall of 0.00 tons, a weighted TDD shortfall of 89.36, a fixed cost of 62, and a variable cost of 80.46. Table 6.11 describes the TDD violations. For customer 0, there were three late deliveries. The first overdue delivery of 66 tons was late by 4.27 hours and the second overdue delivery of 81 tons was late by 1.40 hours. The weighted penalties were 28.18 and 11.34, respectively. The weighted penalties are determined by *lateDeliveryAmount*hoursLate/10*. This allows the tabu search to differentiate between degrees of lateness.

Table 6.10 AF MTMS Evaluation Summary

Total Demand Shortfall	0
Sum TDD Shortfall (weighted)	89.36
Fixed Cost	62.00
Variable Cost	80.46
MOG Parking Penalty	0
Total cost	**231.82**

Table 6.11 TDD Shortfall Details

Customer	Late Demand	Length of Time	Weighted shortfall
0	66	4.27	28.18
0	81	1.40	11.34
0	84	0.36	3.02
1	33	2.65	8.75
1	4	1.94	0.78
2	12	0.21	0.25
2	1	3.69	0.37
3	42	2.08	8.74
3	45	4.07	18.32
4	16	4.90	7.84
5	8	2.13	1.70
6	56	0.02	0.11
Total (weighted)			**89.36**

Tables 6.12 and 6.13 are examples of the solution output presented in Appendix A. In Table 6.12, vehicle 5's tour consists of five trips. As specified in the initial data, vehicle 5 is first available at time 0.00 to begin its tour. It begins its on-load at the depot at time 0.0 where it spends no time in the load queue. The vehicle requires 4 hours to load, and departs when it completes loading. The vehicle arrives at customer 1 at time 4.85 and immediately begins unloading 85 tons of goods. The unloading process takes 2

hours and the vehicle departs as soon as unloading is complete. The vehicle returns to the depot by time 7.71. The vehicle then requires 2 hours to service before it begins on-loading at time 9.71. The vehicle's second trip services customer 2. The vehicle returns to the depot at time 17.42 and requires 2 hours to service. The vehicle's final trip services two customers. It delivers 36 tons to customer 5 and 49 tons to customer 1. The vehicle completes its tour upon arrival at the depot at time 48.8. Notice that the delivery to customer 0 was at time 26.27. This is a late delivery of 66 tons by 4.27 hours, which is documented in Table 6.11. Customer 1 also received a 1.94-hour late delivery of 4 tons and customer 6 received a 0.02-hour late delivery of 56 tons.

Table 6.12 Example of a Vehicle Tour

Vehicle	tour	Arr Time	Start Off/Onload	End Off/Onload	Depart	Delivery
5	Depot	0	0	4	4	
5	1	4.85	4.85	6.85	6.85	85
5	Depot	7.71	9.71	13.71	13.71	
5	2	14.56	14.56	16.56	16.56	85
5	Depot	17.42	19.42	23.42	23.42	
5	0	24.27	24.27	26.27	26.27	85
5	Depot	27.12	29.12	33.12	33.12	
5	6	34.02	34.02	36.02	36.02	84
5	Depot	36.91	38.91	42.91	42.91	
5	5	43.81	43.81	45.81	45.81	36
5	1	45.94	45.94	47.94	47.94	49
5	Depot	48.8				

Table 6.13 is a detailed example of a customer's delivery schedule. This table illustrates the vehicle letter, customer service letter, and vehicle trip letter used for the delivery. It also indicates how the delivery schedule adheres to some of the constraints. For example, the ETDD constraint was set to 12 and the working MOGA was set to 1. Notice that two vehicles arrived at customer 0 prior to time 12 and they were not allowed

to unload until time 12. The customer also formed a FIFO queue to unload the goods of the first three vehicles. Deliveries 6, 7 and 8, in bold, also waited in a queue to unload their goods.

Table 6.13 Example of a Customer Delivery Schedule

Customer	CustLet	Veh	VehLet	Delivery	Arr	Off-load	Dep
0	70	11	41	12	7.21	12	13
0	73	12	46	12	7.21	13	14
0	86	10	37	12	13.77	14	15
0	82	16	66	12	15.78	15.78	16.78
0	76	15	62	12	16.92	16.92	17.92
0	80	7	23	12	**17.35**	**17.92**	18.92
0	72	6	18	12	**17.78**	**18.92**	19.92
0	74	8	28	12	**18.99**	**19.92**	20.92
0	79	13	53	12	21	21	22
0	71	5	13	85	24.27	24.27	26.27
0	83	3	8	85	34.4	34.4	36.4
0	88	15	65	12	41.03	41.03	42.03
0	77	16	69	6	41.88	42.03	43.03
0	75	2	4	84	44.36	44.36	46.36

The total run time for 10,000 iterations on a Pentium III was 122 minutes, where the best-found solution was generated at iteration 4,605 in 51 minutes. However, some satisfactory solutions were found early in the tabu search process. Table 6.14 provides examples of when some of these solutions were found. For these instances, all the demands were filled.

Table 6.14 AF MTMS Example Solutions

Demand Shortfall	TDD Shortfall	Fixed Cost	Variable Cost	Total Cost	Iteration	Time
0.00	92.96	62	81.02	235.98	3151	33.65
0.00	95.85	62	80.18	238.04	1750	17.93
0.00	98.56	62	80.67	241.24	1203	11.79
0.00	102.71	62	80.11	245.82	754	7.14

Figure 6.5 presents tabu search returns over time. Each point in the chart were the new best total cost solutions found as the search progressed. The search made significant improvements from the initial starting solution in the first 200 iterations. Satisfactory solutions were found around the 800th iteration. The margin of improvement diminished after the 1,000th iteration.

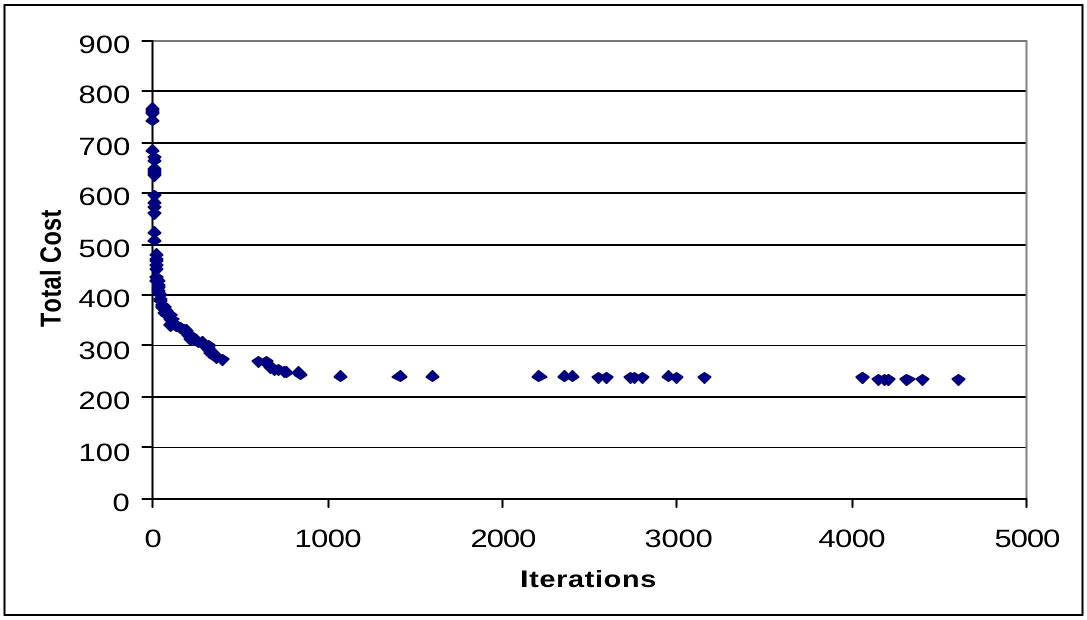

Figure 6.5 Tabu Search Solutions Over Time (Iterations)

Figure 6.6 displays the tabu search objective function values for the first 5,000 iterations. The four lines are the total cost, fixed + variable (F+V) cost, TDD shortfall, and demand shortfall. The chart displays a "healthy" tabu search process that moves through different areas of the solution space. This is evident by the variation of the objective function values. The search also reveals no evidence of cycling.

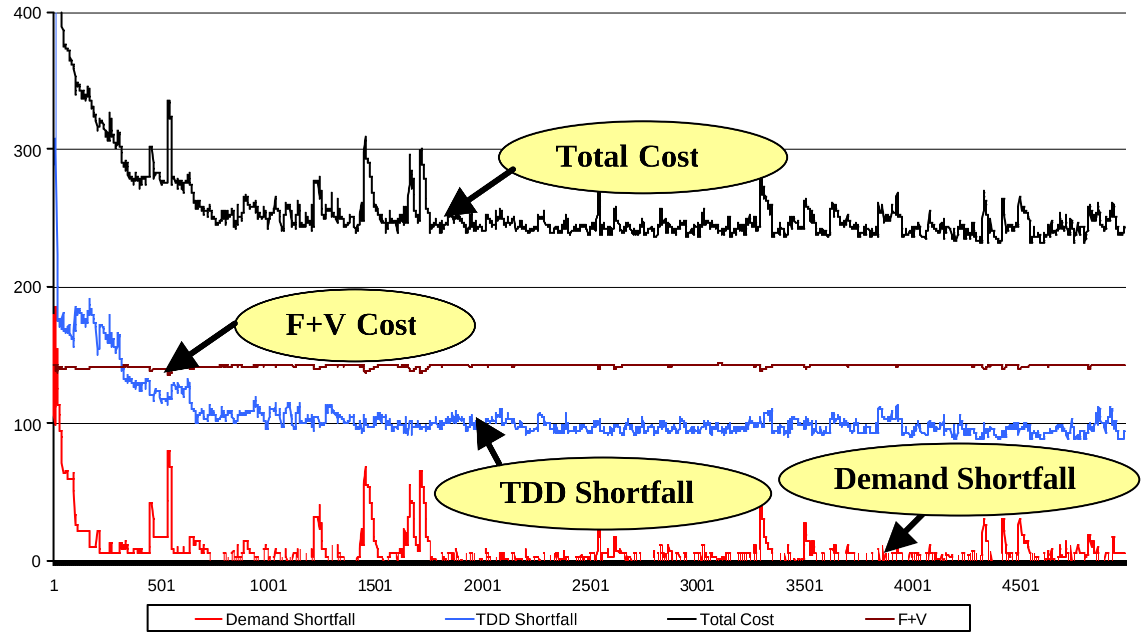

Figure 6.6 GTTS Search Pattern

6.2.3 Summary

The GTTS algorithm successfully found good solutions for the Air Force MTMS instance. The algorithm found a number of solutions that satisfied all customer demands and with minimal late delivery times. The final solution details are displayed in Appendix A, where vehicle loads and schedules and customer deliveries are detailed. Air planners can use this model to determine how to schedule aircraft to best meet customer requirements.

6.3 Joint MTMS TDVRSP Instance

There are twelve MTMS instances in the benchmark data set that model both cargo aircraft and ground vehicles. The instances are multiple trip multiple service, multiple depot, multiple nonhomogeneous vehicle TDVRSP instances with direct delivery vehicles, working and parking MOG constraints, and time window constraints. In this section, benchmark data set 32 and its solution are presented as an example of the joint MTMS instance. Figure 6.7 presents the APOD, SPOD and customer locations relative to each other. The arrow pointing from coordinate (0,0) is the reference point for the direct delivery vehicles. The APOD is approximately 400-500 miles from each customer location. The SPOD is approximately 200-300 miles from each customer. The customers are Corps and Division SSAs. Customers 0, 1, 2, and 3 can accommodate Air Force cargo aircraft, whereas the others cannot. All aircraft are located at the APOD and all ground vehicles are located at the SPOD.

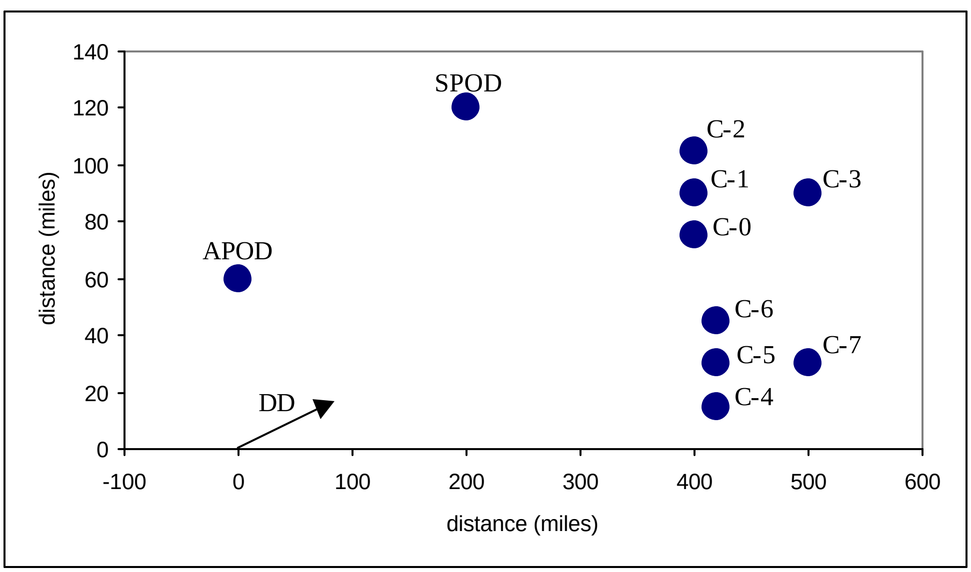

Figure 6.7 Joint MTMS APOD, SPOD and Customer Distribution

6.3.1 Data and parameter settings

Tables 6.15, 6.16, and 6.17 display customer data used in the model. There are 8 customers with varying characteristics. Demands range from 240 to 520 tons. The working and parking MOGs vary in size from 1–3 for the working MOGA and MOGG and 2 – 6 for the parking MOGA. All customers have ETDD constraints = 0, while customers 1, 4, and 6 have time windows that restrict arrival and departure times. Table 6.17 presents multiple TDD requirements for each customer. In Table 6.17, the demand is displayed as cumulative demand, where those cumulative demand levels are required for each specified TDD requirement. Actual delivery levels less than those specified are weighted and penalized. Throughout the model, the APOD is restricted to a working MOGA = 4 and the SPOD has a working MOGG = 4.

Table 6.15 Joint MTMS Customer Data

Cust ID	locationCoor		Demand	wMOGA	wMOGG	pMOGA	ETDD	TDD	Priority
0	400	75	490	1	3	2	0	46	1
1	400	90	490	1	1	2	0	48	1
2	400	105	500	2	2	4	0	42	1
3	500	90	520	3	1	6	0	48	1
4	420	15	250	0	1	0	0	48	1
5	420	30	240	0	2	0	0	46	1
6	420	45	260	0	3	0	0	44	1
7	500	30	250	0	1	0	0	48	1

Table 6.16 Joint MTMS Customer Time Windows

Customer ID	AC-time windows		G-time windows	
1	48	60	48	60
4	24	36	24	36
6	6	18	6	18

Table 6.17 Joint MTMS Customer TDD

cust ID	cummulative demand	ETDD	TDD
0	102	0	10
0	244	0	22
0	374	0	35
0	490	0	46
1	125	0	14
1	251	0	22
1	375	0	34
1	510	0	48
2	99	0	18
2	262	0	23
2	364	0	32
2	500	0	42
3	82	0	14
3	275	0	21
3	373	0	33
3	520	0	48
4	88	0	16
4	122	0	24
4	250	0	48
5	50	0	12
5	110	0	20
5	180	0	30
5	270	0	46
6	144	0	24
6	192	0	36
6	270	0	44
7	95	0	14
7	155	0	24
7	190	0	36
7	270	0	48

Vehicle data used in the Joint MTMS instance is displayed in Table 6.18. For this instance, there are two aircraft types and five ground vehicle types. The aircraft closely resemble the C-17 and C-130 cargo aircraft. Vehicles 0 and 1 are direct delivery vehicles that originate outside the theater of operations and their available times are greater than 0. Those same vehicles later become theater assets and are represented as vehicles 2 and 4. The ground vehicles represent heavy and medium fleet assets. The capacity is in tons,

speed is in mph, and service, load, and unload times are in hours. The available time is in hours relative to the model start time. The cruising length is in miles and is not constraining in this problem. The fixed costs and variable costs are notional and are used simply as a means to minimize the use of aircraft and distances traveled.

Table 6.18 Joint MTMS Vehicle Data

Veh	# Trips	Cap	Speed	Serv Time	Location Coordinates		Fixed Cost	Var Cost	AorG	Avail Time	CL	DD	Load Time	Unload Time
0	1	85	470	2	0	0	0	0.05	A	10	9000	1	4	2
1	1	85	470	2	0	0	0	0.05	A	20	9000	1	4	2
2	3	85	470	2	0	60	10	0.05	A	20	9000	0	4	2
3	2	85	470	2	0	60	10	0.05	A	30	9000	0	4	2
4	4	85	470	2	0	60	10	0.05	A	0	9000	0	4	2
5	5	12	330	1	0	60	2	0.02	A	0	3000	0	2	1
6	5	12	330	1	0	60	2	0.02	A	0	3000	0	2	1
7	5	12	330	1	0	60	2	0.02	A	0	3000	0	2	1
8	5	12	330	1	0	60	2	0.02	A	0	3000	0	2	1
9	5	12	330	1	0	60	2	0.02	A	0	3000	0	2	1
10	5	12	330	1	0	60	2	0.02	A	0	3000	0	2	1
11	5	12	330	1	0	60	2	0.02	A	12	3000	0	2	1
12	5	12	330	1	0	60	2	0.02	A	12	3000	0	2	1
13	5	12	330	1	0	60	2	0.02	A	12	3000	0	2	1
14	5	12	330	1	0	60	2	0.02	A	12	3000	0	2	1
15	5	12	330	1	0	60	2	0.02	A	12	3000	0	2	1
16	3	28	45	2	200	120	2	0.01	G	0	300	0	3	1
17	3	28	45	2	200	120	2	0.01	G	0	300	0	3	1
18	3	28	45	2	200	120	2	0.01	G	0	300	0	3	1
19	3	28	45	2	200	120	2	0.01	G	0	300	0	3	1
20	3	28	45	2	200	120	2	0.01	G	0	300	0	3	1
21	3	28	45	2	200	120	2	0.01	G	6	300	0	3	1
22	3	28	45	2	200	120	2	0.01	G	6	300	0	3	1
23	4	16	60	1	200	120	1	0.01	G	6	300	0	2	1
24	4	16	60	1	200	120	1	0.01	G	0	300	0	2	1
25	4	16	60	1	200	120	1	0.01	G	0	300	0	2	1
26	4	16	60	1	200	120	1	0.01	G	0	300	0	2	1
27	4	16	60	1	200	120	1	0.01	G	8	300	0	2	1
28	4	16	60	1	200	120	1	0.01	G	8	300	0	2	1
29	4	16	60	1	200	120	1	0.01	G	8	300	0	2	1
30	4	12	60	1	200	120	1	0.01	G	0	300	0	1	1
31	4	12	60	1	200	120	1	0.01	G	0	300	0	1	1
32	4	12	60	1	200	120	1	0.01	G	0	300	0	1	1
33	4	12	60	1	200	120	1	0.01	G	8	300	0	1	1
34	4	12	60	1	200	120	1	0.01	G	8	300	0	1	1
35	4	12	60	1	200	120	1	0.01	G	8	300	0	1	1
36	3	8	45	1	200	120	1	0.01	G	0	300	0	1	1
37	4	4	60	1	200	120	1	0.01	G	0	300	0	1	1
38	4	4	60	1	200	120	1	0.01	G	0	300	0	1	1
39	4	4	60	1	200	120	1	0.01	G	12	300	0	1	1
40	4	4	60	1	200	120	1	0.01	G	12	300	0	1	1

Tables 6.19 and 6.20 provide the vehicle trip letters and customer service letters used to format the TDVRSP in S_n. There are 157 vehicle trip letters and 195 customer service letters for a total of 352 letters.

Table 6.19 Vehicle Trip Letters

Vehicle	Vehicle Trip Letters	Vehicle	Vehicle Trip Letters
0	0	21	81,82,83
1	1	22	84,85,86
2	2,3,4	23	87,88,89,90
3	5,6	24	91,92,93,94
4	7,8,9,10	25	95,96,97,98
5	11,12,13,14,15	26	99,100,101,102
6	16,17,18,19,20	27	103,104,105,106
7	21,22,23,24,25	28	107,108,109,110
8	26,27,28,29,30	29	111,112,113,114
9	31,32,33,34,35	30	115,116,117,118
10	36,37,38,39,40	31	119,120,121,122
11	41,42,43,44,45	32	123,124,125,126
12	46,47,48,49,50	33	127,128,129,130
13	51,52,53,54,55	34	131,132,133,134
14	56,57,58,59,60	35	135,136,137,138
15	61,62,63,64,65	36	139,140,141
16	66,67,68	37	142,143,144,145
17	69,70,71	38	146,147,148,149
18	72,73,74	39	150,151,152,153
19	75,76,77	40	154,155,156,157
20	78,79,80		

Table 6.20 Customer Service Letters

Customers	Service letters
0	158-187
1	188-217
2	218-247
3	248-272
4	273-292
5	293-312
6	313-332
7	333-352

The GTTS parameters used to solve this particular Joint MTMS problem are displayed in Table 6.21 and justified in Chapter 8. For this problem, there were a total of 9,000 tabu search iterations. The load distribution option was set at 1. Option 1 is when vehicle ordering determines the amount of service for each customer. The number of worsening moves and constant move tolerances were set at 3 and 5 for normal tabu search iterations and intensification tabu search iterations. The super diversification parameters are set to exercise 6 diversification moves when 20 total cost solution values are within ±0.01% of each other within a block of 200 solutions. The *allow redundant moves* parameter is turned on, which allows the search to select the identity permutation within the evaluated orbit. The route length was turned off for this instance. The storage constraint is turned off because there are no hubs with constraining storage facilities in this instance. An elite list of size 1 was used as the incumbent solution for the intensification iterations. The move tabu tenure is 3, while the orbit tabu tenure is indefinite. The conjugacy class tabu list was not turned on.

Table 6.21 Joint MTMS GTTS Parameters

Parameter	Parameter setting
periodLength	48
sizeGroups	5
neighborhoodSizeLimit	500
dataSet	32
iterations	250
intensificationIterations	50
maxLoops	30
loadDistributionOption	1
moveTabuTenure	3
worseningMoveTolerance	3
constantMoveTolerance	5
intensificationWorseningMoveTolerance	3
intensificationConstantMoveTolerance	5
eliteListSize	1
superDiversifyRange	200
superDiversifyTolerance	20
superDiversifyMoves	6
objFunctionWeights	{5,1,1, .05}
allowConjugate Tabu List	FALSE
allowRedundantMoves	TRUE
allowRedundantMovesIntensification	TRUE
earlyTDDIsHard	FALSE
storageConstraint	FALSE
routeLengthConstraint	FALSE
fuelResupplyAvailable	FALSE
timeWindowConstraint	TRUE
allowMOGConstraint	TRUE
allowPUMOGConstraint	TRUE

6.3.2 Results

The Joint MTMS results are summarized in Table 6.22. Detailed vehicle loads, schedules, and customer deliveries are displayed in Appendix B, Tables B.1 and B.2. The solution, in terms of S_n, is displayed below. The conjugacy class of the solution is $2^{151}3^{6}4^{1}$. Most of the vehicle trips had single customer deliveries and some had multiple customers.

(0 251)(1 252)(2 170)(3 225)(4 257)(5 249)(6 215)(7 250)(8 189)
(9 219)(10 174)(11 175)(12 187)(13 182)(14 247)(15 237)(16 202 188)
(17 230)(18 260)(19 164)(20 204)(21 177)(22 191)(23 222)(24 241)
(25 171)(26 216)(27 192)(28 184)(29 227)(30 234)(31 217)(32 232)
(33 224)(34 209)(35 165)(36 180)(37 194)(38 195)(39 244)(40 176)
(41 181)(42 196)(43 226)(44 236)(45 163)(46 220)(47 159)(48 228)
(49 162)(50 265)(51 242)(52 238)(53 253)(54 207)(55 214)(56 198)
(57 173)(58 229)(59 169)(60 172)(61 200)(62 239)(63 272)(64 166)
(65 259)(66 201)(67 318)(68 351)(69 161)(70 319)(71 235)(72 346)
(73 320)(74 348)(75 278)(76 321)(77 324)(78 243)(79 331)(80 311)
(81 285)(82 206)(83 273)(84 284)(85 205)(86 281)(87 350)(88 294)
(89 275)(90 280)(91 168)(92 335)(93 178)(94 158)(95 333)(96 326)
(97 160)(98 269)(99 310)(100 290)(101 203)(102 288)(103 212)
(104 218)(105 332)(106 349 315)(107 309)(108 179)(109 193)
(110 274 352)(111 342)(112 299)(113 231)(114 312)(115 292)(116 223)
(117 339)(118 314)(119 293)(120 341)(121 337)(122 254)(123 298)
(124 295)(125 334)(126 317)(127 296)(128 304)(129 330)(130 211 327)
(131 297)(132 305)(133 343)(134 316)(135 336)(136 306)(137 301)
(138 276)(139 300)(140 338)(141 302)(142 344)(143 261)(144 197)
(145 329)(146 221 190)(147 289)(148 233)(149 240)(150 340)(151 308)
(152 279)(153 266 185 199)(154 328)(155 307)(156 255)(157 313 246)

For the best-found solution, there was a demand shortfall of 0.0 tons, a weighted TDD shortfall of 414.39, a fixed cost of 84, and a variable cost of 92.62. There were no parking MOG violations in the solution. TDD violation details are located in Appendix B, Table B.3. The magnitude of TDD violations is not surprising for this problem due to the instance's high time requirement density. For example, customer 3 received five of

the eleven C-17 deliveries and still did not have all its TDD requirements' satisfied. Table 6.23 is the TDD shortfall violations for customer 3, and Table 6.24 displays customer 3's delivery schedule.

Table 6.22 Joint MTMS Evaluation Summary

Total Demand Shortfall	0
Sum TDD Shortfall (weighted)	414.39
Fixed Cost	84.00
Variable Cost	92.62
MOG Parking Penalty	0
Total cost	**591.01**

Table 6.23 Customer 3 TDD Shortfall Summary

Customer ID	Late demand	Length of time	Weighted shortfall
3	85	2.08	17.68
3	4	6.21	2.48
3	4	9.98	3.99
3	12	1.43	1.72
3	78	4.06	31.67
3	11	0.89	0.98
3	16	1	1.60
3	12	2	2.40
Total (weighted)			**62.52**

Table 6.24 Customer 3 Delivery Schedule

Customer	CustLet	Veh	VehLet	Delivery	Arr	Off-load	Dep
3	250	4	7	85	5.07	5.07	7.07
3	251	0	0	85	11.08	11.08	13.08
3	260	6	18	12	19.53	19.53	20.53
3	252	1	1	85	21.08	21.08	23.08
3	261	37	143	4	26.21	26.21	27.21
3	253	13	53	12	29.99	29.99	30.99
3	272	15	63	12	33.43	33.43	34.43
3	249	3	5	85	35.07	35.07	37.07
3	255	40	156	4	40.97	40.97	41.97
3	257	2	4	85	45.68	45.68	47.68
3	265	12	50	12	45.88	45.88	46.88
3	259	15	65	11	47.89	47.89	48.89
3	269	25	98	16	48.01	48.01	49.01
3	254	31	122	12	48.87	49.01	50.01

The total run time for 9,000 iterations on a Pentium III was 434.69 minutes, where the best-found solution was found at iteration 8976 (433 minutes). However, some satisfactory solutions were found earlier in the tabu search process. Table 6.25 provides solution examples of satisfactory solutions. For these instances, all the demands were filled.

Table 6.25 Example Solutions

Demand Shortfall	TDD Shortfall	Fixed Cost	Variable Cost	Total Cost	Iteration	Time
0	423.02	84	92.92	599.94	3850	180.59
0	435.13	84	92.95	612.09	2701	123.54
0	444.30	84	92.75	621.06	1891	85.15
0	459.14	84	92.93	636.07	851	37.14
0	521.00	84	92.68	697.69	446	20.07

6.4 Conclusion

The MTMS primarily differentiates itself from the typical GVRP with the addition of multiple vehicle trips, multiple customer services, and MOG constraints. The MTMS instance also includes many of the typical GVRP dimensions and constraints. They are single or multiple depots, single or multiple homogeneous or multiple nonhomogeneous vehicles, time window constraints, and route length constraints. Time window constraints are in the form of hard ETDD, soft TDD, and hard no delivery time windows. The MTMS primary objectives are to minimize the amount of unmet demand, late delivery, vehicle fixed costs, and vehicle variable costs.

In this chapter, two MTMS instances were presented and solved. The Air Force MTMS consisted exclusively of aircraft as vehicles. The Joint MTMS TDVRSP consisted of air and ground vehicles. Each instance was a multiple trip, multiple service, multiple nonhomogeneous vehicle TDVRSP with time windows and MOGs. The Air Force MTMS had a single depot, while the Joint MTMS had multiple depots. Each instance was solved using group theoretic tabu search. Results were satisfactory and details for each solution are provided in Appendices A and B.

VII. The Multiple Trips Multiple Services with Hub TDVRSP Instance

7.1 Introduction

The multiple trips multiple services with hub (MTMS with hub) TDVRSP is one of the theater distribution hierarchy instances discussed in Section 1.4.2. This problem is applicable to theater distribution logistics problems with a hierarchy of delivery tiers. In this chapter, the general MTMS with hub instance is defined and an example is presented.

7.2 MTMS with Hub General Concept

The MTMS with hub instance has the same dimensions and constraints as the MTMS instance except for the addition of hubs and storage constraints. Hubs are defined as transshipment nodes that receive, store, and distribute cargo within the theater distribution network.

In terms previously defined, a hub is both a customer and a depot. It is a customer in the sense it demands specific amounts of cargo in accordance with prescribed time requirements. It is a depot in the sense that it is the supply source of cargo for customers. A hub has vehicles assigned to its location that deliver logistics to customers within the hub's area of responsibility.

Hubs are constrained similarly to customers in the MTMS instance. They may have time windows, working MOG and parking MOGs, and ETDDs. Additionally, hubs may have storage constraints. Storage constraints are defined as the total amount of cargo a hub may have on hand at any given time. Storage constraint violations are soft constraints and are penalized in the objective function. The storage constraint penalty

used in the objective function is presented in Section 4.2.2.2, where each hub is penalized for the amount of logistics on hand greater than the maximum storage amount.

The MTMS with hub has a tiered distribution architecture. The first order tier contains the depots and customers/hubs served by the depots. Middle tiers consist of hubs that service customers/hubs. The last order tiers consist of end customers served by a hub. Each tier is a self-contained distribution network. However, they are not independent of each other. Lower ordered tiers are dependent on higher ordered tiers. For example, the hubs in a lower ordered tier receive logistics as a customer within a higher ordered tier. Once a hub receives its supply, it can distribute cargo to its customers.

Figure 7.1 is an example of a MTMS with hub instance. There are four tiers within this network. Tier 0 is the APOD with its customers (1,2,CSA). Tier 1 is the CSA as a hub and its customers (DSA1, DSA2, 3). Tier 2 is DSA1 as the hub and its customers (BSA3, BSA4, and 5). Tier 3 is DSA2 as the hub and its customers (BSA1, BSA2, and 4).

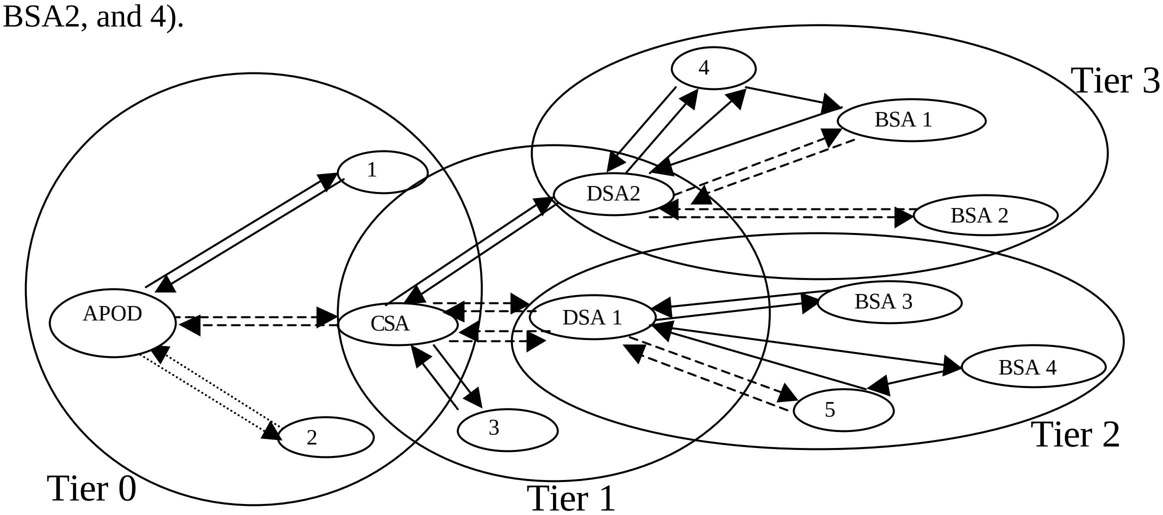

Figure 7.1 MTMS with hub Instance

The hub is allowed to distribute its cargo after it is received and processed. Cargo is received as supply objects, which are characterized with the amount of demand delivered and time of delivery. Each supply object is processed and prepared for delivery to its next customer. Processing times may be based on the amount of supply. For this research, a processing time of 1 hour was used for all supply objects. Supply objects are either loaded directly onto available vehicles or stored for later delivery.

The MTMS with hub instance has the same assumptions as the MTMS instance, plus a few more that account for the hub dimension. Additional assumptions are listed below.

1. All hubs have no supply in storage at time 0.
2. Vehicles depart and return to the same hub*.
3. Vehicles wait to depart hubs with full loads, unless customer demands require less.
4. Customers within the same tier are serviced exclusively from a single hub*.
5. For a hub, working MOGs that unload vehicles are independent of working MOGs that load vehicles*.

Like the MTMS instance assumptions, these assumptions can be removed by coding additional methods within the GTTS TDVRSP. Assumptions labeled with an * require more extensive coding for removal. However, as the number of methods within the algorithm increase, so does the time required to obtain a solution.

7.3 MTMS with Hub Instance

There are fifteen MTMS with hub instances in the benchmark problems that model multi-modal vehicles. The instances are multiple trips multiple services, multiple depot, multiple nonhomogeneous vehicle TDVRSP instances with direct delivery vehicles, working and parking MOG constraints, route length constraints, time window constraints, and storage constraints. In this section, benchmark data set 37 and its solution are presented as an example of the MTMS with hub instance. Figure 7.2 presents the depot, hub, and customer locations relative to each other. Customers 0, 1, 2, and 3 can accommodate cargo aircraft, whereas the others cannot. All aircraft are located at the APOD. Table 7.1 presents the tier structure for the MTMS with hub instance.

Table 7.1 MTMS with hub Tier Structure

Tier	Supply source	Customers
0	APOD, SPOD	C-0, C-1, C-2, C-3
1	C-2	C-4, C-5, C-6, C-7
2	C-3	C-8, C-9, C-10, C-11, C-12
3	C-10	C-13, C-14, C-15, C-16, C-17, C-18
4	C-11	C-19, C-20, C-21, C-22, C-23, C-24
5	C-12	C-25, C-26, C-27, C-28, C-29, C-30

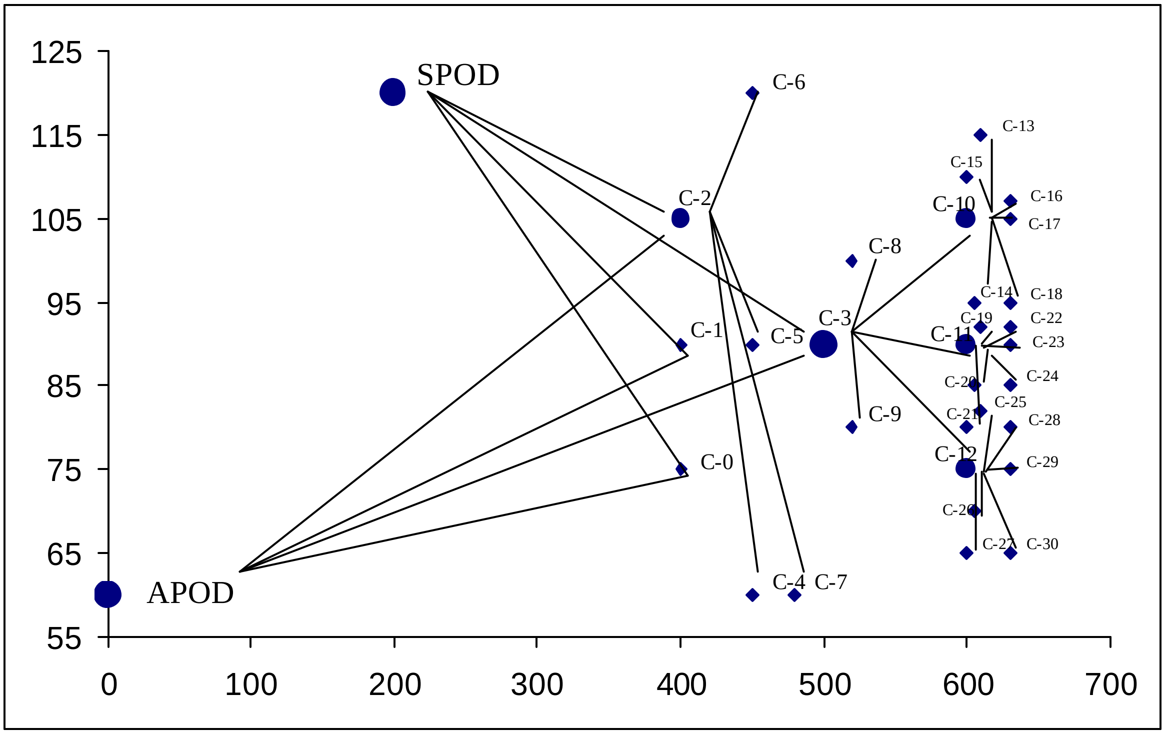

Figure 7.2 Depot, Hub, and Customer Locations

Tables 7.2, 7.3, and 7.4 display customer data used in the model. There are a total of 31 customers with varying characteristics. Demands range from 65 to 1800 tons. The working and parking MOGs vary in size from 1–3 for the working MOGA and MOGG and 2 – 6 for the parking MOGA. The APOD is restricted to a working MOGA = 6 and the SPOD has a working MOGG = 8. Some customers have ETDD constraints > 0 and some have time windows that restrict arrival and departure times. Table 7.3 presents TDD requirements for the customers with more than one TDD. In Table 7.4, time windows are displayed for the APOD and customer 2, which restrict aircraft arrival or departure within those windows.

Table 7.2 MTMS with Hub Customer Data

Cust	Location Coordinates		Demand	wMOGA,wMOGG,pMOGA			eTDD	TDD	Hub	Tier	Prior	Storage	ACfuel	Gfuel
0	400	75	300	1	3	2	0	96	0	0	1	0	0	200
1	400	90	300	2	3	4	0	96	0	0	1	0	0	200
2	400	105	800	3	3	6	0	96	1	0	1	200	2000	400
3	500	90	1800	3	3	6	0	96	2	0	1	300	8000	500
4	450	60	200	0	2	0	0	96	0	1	1	0	0	200
5	450	90	200	0	2	0	0	96	0	1	1	0	0	200
6	450	120	200	0	1	0	36	96	0	1	1	0	0	200
7	480	60	200	0	1	0	24	96	0	1	1	0	0	200
8	520	100	75	0	1	0	0	48	0	2	1	0	0	50
9	520	80	75	0	2	0	0	72	0	2	1	0	0	50
10	600	105	550	0	3	0	0	96	3	2	1	0	0	200
11	600	90	550	0	3	0	0	96	4	2	1	0	0	200
12	600	75	550	0	3	0	0	96	5	2	1	0	0	200
13	610	115	65	0	1	0	0	24	0	3	1	0	0	50
14	605	95	70	0	2	0	0	36	0	3	1	0	0	50
15	600	110	70	0	2	0	0	72	0	3	1	0	0	50
16	630	107	115	0	3	0	0	30	0	3	1	0	0	50
17	630	105	115	0	3	0	0	66	0	3	1	0	0	50
18	630	95	115	0	3	0	0	96	0	3	1	0	0	50
19	610	92	65	0	1	0	24	70	0	4	1	0	0	50
20	605	85	70	0	2	0	24	96	0	4	1	0	0	50
21	600	80	70	0	3	0	24	96	0	4	1	0	0	50
22	630	92	115	0	1	0	24	96	0	4	1	0	0	50
23	630	90	115	0	3	0	24	36	0	4	1	0	0	50
24	630	85	115	0	3	0	24	90	0	4	1	0	0	50
25	610	82	65	0	1	0	48	96	0	5	1	0	0	50
26	605	70	70	0	2	0	48	96	0	5	1	0	0	50
27	600	65	70	0	3	0	48	96	0	5	1	0	0	50
28	630	80	115	0	2	0	48	72	0	5	1	0	0	50
29	630	75	115	0	3	0	48	96	0	5	1	0	0	50
30	630	65	115	0	3	0	48	72	0	5	1	0	0	50

Table 7.3 Customers with Multiple TDD Requirements

Cust	Cumulative Demand	ETDD	TDD
0	69	0	36
0	142	0	60
0	219	0	84
0	300	0	96
1	70	0	36
1	150	0	60
1	215	0	84
1	300	0	96
4	90	0	36
4	200	0	96
5	110	0	60
5	200	0	96
7	80	24	60
7	200	24	96

Table 7.4 Customer Time Windows

Cust	AC-Time windows		G- Time windows	
APOD	24	30	0	0
APOD	48	54	0	0
APOD	72	78	0	0
2	24	30	0	0
2	48	54	0	0
2	72	78	0	0

The vehicle data utilized in the MTMS with hub instance is displayed in Appendix D. There are 90 vehicles, where each vehicle has its own identification number and characteristics. For this instance, there are two aircraft types and five ground vehicle types. The aircraft closely resemble the C-17 and C-130 cargo aircraft. Vehicles 0-4 are the C-17 representatives and vehicles 5-11 are the C-130 representatives. Vehicles 0 and 1 are direct delivery vehicles that originate outside the theater of operations and their available times are greater than 0. The ground vehicles represent heavy and medium fleet assets. The capacity is in tons, speed is in mph, and service, load, and unload times are in hours. The available time is in hours relative to the model start time. The cruising length is in miles. The fixed costs and variable costs are hypothetical and are used simply as a means to minimize the use of aircraft and distances traveled.

Tables 7.5 and 7.6 provide the vehicle trip letters and customer service letters used to format the TDVRSP in S_n. There are 597 vehicle trip letters and 559 customer service letters for a total of 1156 letters. There are fewer customer service letters than vehicle trip letters because the customer demand is less than total vehicle capacity.

Table 7.5 Vehicle Trip Letters

Vehicle	Vehicle Trip Letters	Vehicle	Vehicle Trip Letters	Vehicle	Vehicle Trip Letters
0	0	30	178-184	60	411-416
1	1	31	185-191	61	417-422
2	2-7	32	192-198	62	423-429
3	8-14	33	199-208	63	430-436
4	15-21	34	209-218	64	437-442
5	22-28	35	219-228	65	443-448
6	29-35	36	229-238	66	449-454
7	36-42	37	239-248	67	455-460
8	43-49	38	249-258	68	461-466
9	50-56	39	259-268	69	467-472
10	57-63	40	269-275	70	473-478
11	64-70	41	276-282	71	479-484
12	71-75	42	283-289	72	485-491
13	76-80	43	290-296	73	492-498
14	81-85	44	297-303	74	499-504
15	86-90	45	304-310	75	505-510
16	91-96	46	311-317	76	511-516
17	97-102	47	318-324	77	517-522
18	103-108	48	325-331	78	523-528
19	109-114	49	332-338	79	529-534
20	115-120	50	339-345	80	535-540
21	121-126	51	346-352	81	541-546
22	127-132	52	353-359	82	547-552
23	133-138	53	360-366	83	553-560
24	139-144	54	367-373	84	561-566
25	145-150	55	374-380	85	567-572
26	151-156	56	381-387	86	573-578
27	157-163	57	388-394	87	579-584
28	164-170	58	395-402	88	585-590
29	171-177	59	403-410	89	591-596

Table 7.6 Customer Service Letters

Customers	Service letters	Customers	Service letters	Customers	Service letters
0	597-615	11	896-935	21	1052-1059
1	616-635	12	936-975	22	1060-1071
2	636-685	13	976-983	23	1072-1083
3	686-765	14	984-991	24	1084-1095
4	766-782	15	992-999	25	1096-1103
5	783-800	16	1000-1011	26	1104-1111
6	801-817	17	1012-1023	27	1112-1119
7	818-835	18	1024-1035	28	1120-1131
8	836-845	19	1036-1043	29	1132-1143
9	846-855	20	1044-1051	30	1144-1155
10	856-895				

The GTTS parameters used to solve this problem are displayed in Table 7.7 and justified in Chapter 8. For this instance, there were a total of 1,500 tabu search iterations. The load distribution option was set at 1. Option 1 is when vehicle ordering determines the amount of service for each customer. The number of worsening moves and constant move tolerances were set at 5 and 8 for normal tabu search iterations and intensification tabu search iterations, respectively. The super diversification parameters are set to exercise 6 diversification moves when 30 total cost solution values ±0.01% are within a block of 200 solutions. The *allow redundant moves* parameter is turned off, which prohibits the search from selecting the identity permutation within the evaluated orbit. The route length and storage constraints were turned on for this instance. An elite list of size 1 was maintained and used as the incumbent solution for the intensification iterations. The move tabu tenure is 3, while the orbit tabu tenure is indefinite. The conjugacy class tabu list was not turned on.

Table 7.7 MTMS with hub GTTS Parameters

Parameter	Parameter setting
periodLength	96
sizeGroups	5
neighborhoodSizeLimit	100
dataSet	37
iterations	350
intensificationIterations	150
maxLoops	3
loadDistributionOption	1
moveTabuTenure	3
worseningMoveTolerance	5
constantMoveTolerance	8
intensificationWorseningMoveTolerance	5
intensificationConstantMoveTolerance	8
eliteListSize	1
superDiversifyRange	200
superDiversifyTolerance	30
superDiversifyMoves	6
objFunctionWeights	{10,1,1, .05}
allowConjugate Tabu List	FALSE
allowRedundantMoves	FALSE
allowRedundantMovesIntensification	FALSE
earlyTDDIsHard	TRUE
storageConstraint	TRUE
routeLengthConstraint	TRUE
fuelResupplyAvailable	TRUE
timeWindowConstraint	TRUE
allowMOGConstraint	TRUE
allowPUMOGConstraint	TRUE

7.4 Results

The MTMS with hub TDVRSP results are summarized in Table 7.8. Detailed vehicle loads, schedules, and customer deliveries are displayed in Appendix C, Tables C.1 and C.2. The solution in terms of S_n is displayed below and is presented to appreciate the magnitude of the problem. The conjugacy class of the solution is $2^{439}3^{6}$.

(0 724)(1 725)(2 687)(3 754)(4 751)(5 738)(6 656)(7)(8 722)(9 721)(10 718)(11 733)(12 657)(13 714)(14)(15 723)(16 688)(17 719)(18 700)(19 760)(20 696 682)(21)(22 624)(23 689

)(24 720)(25 735)(26 693)(27)(28)(29 727)(30 690)(31 755)(32 736)(33 694)(34)(35)(36 728)(37 691)(38 756)(39 703)(40 763)(41)(42)(43 627)(44 692)(45 757)(46 704)(47 730)(48)(49)(50 764)(51 761)(52 758)(53 739)(54 629)(55)(56)(57 731)(58 762)(59 759)(60 740)(61 697)(62)(63)(64 732)(65 695)(66 726)(67 741)(68 665)(69)(70)(71 667)(72 662)(73 659)(74 674)(75 598)(76 599)(77 663)(78 660)(79 607)(80 597)(81 631)(82 630)(83 617)(84 671)(85 600)(86 666)(87 633)(88 618)(89 672)(90 601)(91 634)(92 650)(93 653)(94 605)(95 602)(96 651)(97 636)(98)(99 620)(100 606)(101 637)(102)(103 603)(104 668)(105 621)(106 675 612)(107 604)(108)(109 638)(110 669)(111 622)(112 608)(113 639)(114)(115)(116 670)(117 623)(118 677)(119 708)(120)(121)(122)(123 658)(124 678)(125)(126)(127 709)(128)(129 625)(130 611)(131 642)(132)(133 676)(134 673)(135 626)(136 646)(137 643)(138)(139 609)(140 640)(141)(142 647)(143 644)(144)(145 610)(146 743)(147 628)(148 648)(149 645)(150)(151 679)(152)(153 664)(154 649)(155)(156)(157)(158)(159)(160 752)(161)(162)(163)(164)(165)(166)(167 753)(168)(169)(170)(171)(172747)(173)(174)(175)(176)(177)(178)(179 748)(180)(181)(182)(183)(184)(185)(186 681)(187)(188 654)(189)(190)(191)(192 685)(193 750)(194)(195 655)(196)(197)(198)(199 819)(200 768)(201 799)(202 774)(203 798)(204 791)(205 816)(206 809)(207 781)(208)(209 792)(210 767)(211 810)(212 775)(213 827)(214 820)(215 803)(216826)(217)(218)(219 793)(220 800)(221 769)(222 776)(223 828)(224821)(225 818)(226 811)(227)(228)(229 794)(230 801)(231 770)(232777)(233 829)(234 822)(235 805)(236 812)(237)(238)(239 795)(240 830)(241 813)(242 778)(243 788)(244 823)(245 806)(246 771)(247)(248)(249 796)(250 831)(251 832)(252 807)(253 789)(254 824)(255 779)(256 814)(257)(258)(259 797)(260 790)(261 773)(262 780)(263 772)(264 825)(265 808)(266 815)(267)(268)(269 869)(270 861)(271 923)(272 887)(273 867)(274 971)(275)(276 870)(277 890)(278 924)(279 888)(280 952)(281 972)(282)(283 899)(284 891)(285 838)(286 917)(287 953)(288 945)(289)(290 900)(291 864)(292 898)(293 918)(294 954)(295 847)(296)(297 901)(298 921)(299 871)(300 919)(301 955)(302 848)(303)(304 930)(305 922)(306 872)(307 920)(308 956)(309 877)(310)(311 875)(312 868)(313 873)(314 937)(315 957)(316 949)(317)(318 876)(319 839)(320 958)(321 938)(322 846)(323 897)(324)(325 933)(326 841)(327 931)(328 939)(329 959)(330 951)(331)(332 906)(333 926)(334 932)(335 940)(336 960)(337)(338)(339 907)(340 857)(341 961)(342 941)(343 849)(344 925 893)(345)(346 908)(347 858)(348962)(349 942)(350 878)(351)(352)(353 909)(354 859)(355 879)(356943)(357 963)(358)(359)(360 910 842)(361 860)(362 880)(363 944)(364 964)(365)(366)(367 855)(368 889)(369 965)(370 973)(371 882)(372)(373)(374 912)(375 862)(376 853)(377 974)(378 966)(379)(380)(381 913)(382 975)(383 883)(384 863)(385 967)(386)(387)(388 914)(389 892)(390 968)(391 948)(392 884)(393)(394)(395 915)(396)(397 885)(398 865)(399 969)(400)(401)(402)(403 916)(404 950)(405 886)(406 837)(407 970)(408)(409)(410)(411 977)(412 1011)(413 1001)(414 1020)(415 995)(416 1034)(417 978)(418 1025)(419 990)(420 994)(421 1032)(422 1035)(423 979)(424 1002)(425 991)(426 1022)(427 1021)(428)(429)(430 1006)(431 1003)(432 989)(433 999)(434)(435)(436)(437 981)(438 1004)(439 1014)(440 1013)(441 1023)(442)(443 1018)(444 1005)(445 1015)(446 1026)(447 993)(448)(449 1007)(450 1009)(451 992)(452 1027)(453 1033)(454)(455 984 986)(456 998)(457 1029)(458 1028)(459 1031)(460)(461 985)(462 987)(463)(464 1017)(465 996)(466)(467 1010)(468 1024)(469)(470)(471)(472)(473 1077)(474 1071)(475 1049)(476 1064)(477 1047)(478 1059)(479 1050)(480 1048)(481 1066)(482 1053)(483 1072)(484 1061)(485 1039)(486 1074)(487 1079)(488 1090)(489 1037)(490)(491)(492 1088)(493 1063)(494 1080)(495 1091)(496)(497)(498)(499 1089)(500 1040)(501 1093)(502 1092)(503 1065)(504)(505 1078)(506 1041)(507 1082)(508 1057)(509 1054)(510)(511 1055)(512 1085)(513 1083)(514 1094)(515 1043)(516)(517 1056)(518 1086)(519 1073)(520 1095)(521 1044)(522)(523 1081)(524 1087)(525 1062)(526 1045)(527 1069)(528)(529 1070)(530 1076)(531 1075)(532 1046)(533 1058)(534)(535 1145)(536 1131)(537 1137)(538 1124)(539 1119)(540 1107)(541 1098)(542 1109)(543 1150)(544 1149)(545 1108)(546 1120)(547 1123)(548 1122)(549 1139)(550 1138)(551 1110)(552)(553)(554 1148)(555 1099)(556 1128)(557 1151)(558 1097 1126)(559)(560)(561 1125)(562 1100)(563 1141)(564 1140)(565 1113)(566)(567 1102)(568 1101)(569 1142)(570 1153)(571)(572)(573 1127)(574 1121)(575 1155)(576 1154)(577 1115)(578)(579 1152)(580 1146)(581 1133)(582 1143)(583 1116)(584)(585 1129)(586 1111)(587 1134)(588 1105)(589 1117)(590)(591 1104)(592 1136)(593 1135)(594 1106)(595 1118)(596)

Table 7.8 MTMS with Hub Evaluation Summary

Total Demand Shortfall	0.00
Sum TDD Shortfall (weighted)	5.78
Fixed Cost	106.00
Variable Cost	111.81
Storage Constraint Penalty	3.00
MOG Parking Penalty	0.00
Total cost	**226.59**

For the very best solution, there was a demand shortfall of 0.0 tons, a weighted TDD shortfall of 5.78, a fixed cost of 106.0, and a variable cost of 111.8. There were no parking MOG violations in the solution and a minor storage constraint violation of 3 tons.

Due to route length constraints, some vehicles refueled at customer locations to have enough fuel to return to their depot/hub. Customers 0, 1, and 2 provided 178.6, 146.3 and 384.3 gallons of fuel to ground vehicles.

The total run time for 1,500 iterations was 193 minutes, where the best-found solution was found at iteration 1,361 (180 minutes). However, some satisfactory solutions were found earlier in the tabu search process. Table 7.9 provides solution examples of satisfactory solutions. For these instances, all the demands were filled.

Table 7.9 Example Solutions

Demand Shortfall	TDD Shortfall	Fixed Cost	Variable Cost	Total Cost	Time	Iteration
0.0	5.78	106	111.99	226.77	137.70	1050
0.0	5.78	107	112.80	228.58	44.20	434
0.0	8.80	108	113.25	233.06	19.85	191

With this problem as an example, the GTTS displayed its ability to satisfactorily solve multiple objectives. The two most heavily weighted objectives are the demand shortfall and TDD shortfall. It is most important to fill all customer demand as well as meet their TDD requirements. Secondary objectives include the ability to deliver goods and services economically. Economic efficiency is the ability to deliver goods with less fixed and variable costs. Table 7.9 is a good example of how the GTTS minimized each objective. The demand shortfall, which is the primary objective, was minimized first. TDD was minimized through time as well as the fixed and variable costs. The fixed cost, represents the number of vehicles. Initially, the solution used all available vehicles. As the search progressed, the number of vehicles in the solution was reduced and the fixed cost decreased from 108.0 to 106.0. The vehicles not used in the solution were vehicles 27, 28, 29, and 30. Although the solution, in terms of S_n, indicates that some of these vehicles visited customer 3, they actually did not make the trip because the route length constraint prevented them from doing so. The route length from the SPOD to customer 3 is 301.5 miles. The cruising length of the vehicles is 300 miles.

Figure 7.3 displays the tabu search objective function values for each of the 1,500 iterations. The four lines are the total cost, TDD shortfall, demand shortfall, and fixed cost. The chart displays a "healthy" tabu search process that moves through different areas of the solution space. This is evident by the variation of the objective function values. Variances of the different objective function values indicate different solutions chosen as incumbent solutions.

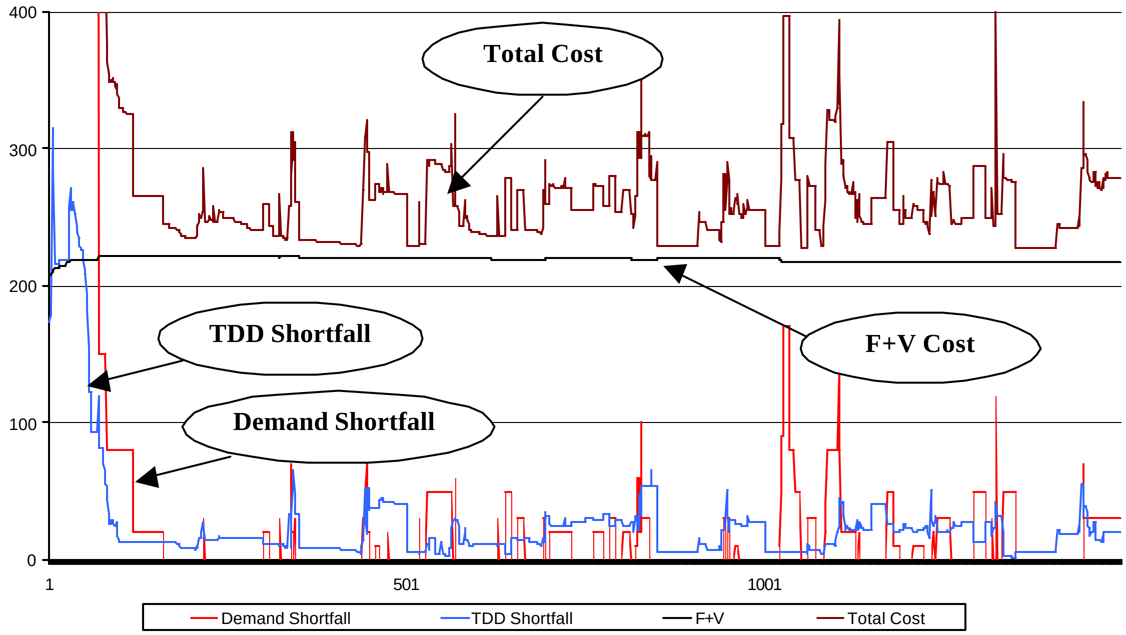

Figure 7.3 MTMS with hub GTTS Graphic

7.5 Summary

The MTMS with hub primarily differentiates itself from the typical GVRP with the addition of multiple vehicle trips, multiple customer services, hubs, MOG constraints, and storage constraints.

The MTMS with hub also includes many of the typical GVRP dimensions and constraints. They are single or multiple depots, single or multiple homogeneous or multiple nonhomogeneous vehicles, time window constraints, and route length constraints. Time window constraints are in the form of hard ETDD, soft TDD, and hard no delivery time windows. The MTMS with hub primary objectives are to minimize the amount of unmet demand, late delivery, vehicle fixed costs, vehicle variable costs, and solution penalties.

In this chapter, one MTMS with hub problem was presented and solved. The problem consisted of 90 air and ground vehicles, two depots, five hubs, and 31

customers. Benchmark problem 37 was used for this example. The problem was a multiple trips, multiple services, multiple nonhomogeneous vehicle TDVRSP with time windows, MOG constraints, storage constraints, and route constraints.

The instance was solved using group theoretic tabu search. Results were satisfactory and details of the solution are provided in Appendix C.

VIII. Computational Results

This chapter documents the results of this research. In Section 8.1, the final solution values for the benchmark problems are presented and a discussion of the general results is given. In Section 8.2, parameter settings for the different problems are detailed. In Section 8.3, the total quality search metrics are used to measure the quality of the algorithm.

8.1 General Results

The GTTS TDVRSP algorithm performed admirably for all the TDVRSP benchmark problem instances. While the algorithm was allowed to execute for significantly longer times (hours) to confirm best-found solutions and observe the models ability to traverse the solution space, the model provided good solutions well within acceptable *practical* time horizons (within minutes) for all problem instances. Summary results are provided in Section 8.1.1. Although not provided in this document, detailed results for each data set are available from the author.

8.1.1 Benchmark Problem Results

The final results for thirty-nine benchmark problems are presented in Tables 8.1, 8.2, and 8.3. Table 8.1 details problems 1 through 9, 28, 31, and 34; Table 8.2 details problems 10 through 18, 29, 32, and 35; and Table 8.3 details problems 19 through 27, 30, 33, 36, 37, 38, and 39.

In these tables, each pair of rows refers to a benchmark problem and each column provides a solution value. The first row of solution values is the *best-found* solution. Best-found solutions are the least total cost solution found during the entire search

process. The second row is from a *good* solution found early in the search process. The solution values displayed are the un-weighted demand shortfall (DS) amount, weighted TDD shortfall amount, vehicle fixed costs (FC), vehicle variable costs (VC), and total costs. For data sets 37, 38, and 39, hub storage penalties are displayed. In addition, the iterations and time required for each solution are noted. The last column (Proc) identifies the computer used. A III indicates a Pentium III with 1.1 Ghz processor and 512 MEG RAM. A IV indicates a Pentium IV with 1.7 Ghz processor and 1024 MEG RAM. For Tables 8.1, 8.2, and 8.3, the model performed 4000, 3000, and 1500 total iterations, respectively.

Discussions of the general results follow the tables in Section 8.1.2. The discussion topics include: processing time for good solutions, problem density effect on solution values, minimizing multiple objectives, use of orbit tabu lists, and the algorithm's run time order of growth.

Table 8.1 Benchmark Problem Group 1

DataSet	Soln	DS	TDD	FC	VC	Total	Iteration	Time	Proc
1	Best	0.00	1.13	54.00	54.28	109.41	2228	19.45	III
1	Good	0.00	3.52	62.00	66.41	131.93	51	0.47	III
2	Best	0.00	0.00	62.00	75.69	137.69	604	8.15	III
2	Good	0.00	0.00	62.00	81.04	143.04	95	1.11	III
3	Best	123.9	0.00	62.00	80.66	266.56	305	3.56	III
3	Good	127.5	0.76	62.00	80.68	270.95	173	1.83	III
4	Best	0.00	3.06	62.00	68.71	133.78	3356	35.34	III
4	Good	0.00	4.80	62.00	70.23	137.04	218	2.13	III
5	Best	0.00	32.62	62.00	81.23	175.85	1202	13.94	III
5	Good	0.00	48.89	62.00	81.65	192.55	218	2.46	III
6	Best	186.6	55.35	62.00	81.32	385.27	3556	45.48	III
6	Good	187.2	118.72	62.00	81.07	449.00	254	2.82	III
7	Best	0.00	3.38	62.00	71.36	136.75	2999	35.21	III
7	Good	0.00	26.28	62.00	77.07	165.36	35	0.40	III
8	Best	0.00	50.62	62.00	81.07	193.71	3977	51.50	III
8	Good	0.00	88.84	62.00	80.96	231.80	222	2.27	III
9	Best	356.7	60.10	62.00	79.14	557.95	3800	38.83	III
9	Good	356.7	89.80	62.00	79.24	587.74	189	1.78	III
28	Best	129.9	0.32	62.00	81.59	273.82	1461	14.17	III
28	Good	131.5	0.32	62.00	81.63	275.45	230	2.02	III
31	Best	0.00	1.71	60.00	67.98	129.69	3823	33.68	III
31	Good	0.00	2.52	62.00	69.07	133.59	157	1.26	III
34	Best	0.00	92.95	62.00	81.02	235.98	3151	33.64	III
34	Good	0.00	102.71	62.00	80.11	244.82	754	7.13	III

Table 8.2 Benchmark Problem Group 2

DataSet	Soln	DS	TDD	FC	VC	Total	Iteration	Time	Proc
10	Best	0.00	0.11	75.00	55.95	131.05	2979	49.65	III
10	Good	0.00	11.89	84.00	60.53	156.43	94	2.30	III
11	Best	0.00	20.16	84.00	89.94	194.11	2406	80.99	III
11	Good	0.00	55.23	84.00	86.96	226.19	743	22.10	III
12	Best	156.5	26.64	84.00	91.82	358.97	2697	98.16	III
12	Good	161.1	40.30	84.00	92.04	377.43	637	20.38	III
13	Best	0.00	20.35	81.00	79.03	180.37	2363	71.69	III
13	Good	0.00	58.18	81.00	73.16	212.34	932	29.64	III
14	Best	0.00	150.23	84.00	92.46	326.70	2951	101.66	III
14	Good	0.00	185.43	84.00	91.78	361.21	758	26.06	III
15	Best	330.8	180.72	84.00	93.63	689.16	2702	109.75	III
15	Good	482	159.86	84.00	85.36	811.22	563	21.38	III
16	Best	0.00	11.99	84.00	78.30	174.29	2403	70.69	III
16	Good	0.00	57.30	84.00	74.05	215.35	597	15.40	III
17	Best	0.00	367.63	84.00	92.60	544.22	2703	88.13	III
17	Good	0.00	402.75	84.00	92.52	579.27	566	17.08	III
18	Best	360	194.33	84.00	92.53	730.86	2701	102.73	III
18	Good	406.2	197.47	84.00	90.18	777.85	620	20.85	III
29	Best	0.00	1.13	80.00	72.64	153.78	2967	153.77	III
29	Good	0.00	3.51	84.00	74.85	162.37	338	9.23	III
32	Best	0.00	438.15	84.00	93.04	615.19	2954	135.70	III
32	Good	0.00	521.00	84.00	92.68	697.69	446	20.07	III
35	Best	271.5	30.33	84.00	92.15	477.98	2703	114.28	III
35	Good	278.00	43.30	84.00	91.50	496.80	428	15.09	III

Table 8.3 Benchmark Problem Group 3

DataSet	Soln	DS	TDD	FC	VC	SC	Total	Iteration	Time	Proc
19	Best	0.00	0.63	107.50	109.26	-	217.39	1420	168.52	IV
19	Good	0.00	6.51	108.00	115.61	-	230.12	112	9.61	IV
20	Best	0.00	7.06	108.00	146.44	-	261.50	1435	163.79	IV
20	Good	0.00	16.50	108.00	147.26	-	271.76	715	64.28	IV
21	Best	837.2	0.61	108.00	150.14	-	1095.95	1218	184.74	IV
21	Good	876.8	3.94	108.00	150.30	-	1139.05	175	12.70	IV
22	Best	0.00	5.39	107.50	111.97	-	224.85	1196	129.21	IV
22	Good	0.00	18.67	108.00	115.38	-	242.06	219	12.83	IV
23	Best	0.00	3.77	108.00	143.59	-	255.36	821	71.22	IV
23	Good	0.00	6.69	108.00	143.77	-	258.47	547	38.27	IV
24	Best	824.4	9.58	108.00	151.45	-	1093.43	1414	170.66	IV
24	Good	839.2	18.58	108.00	150.75	-	1116.53	341	24.16	IV
25	Best	0.00	5.92	106.50	111.58	-	224.01	1418	153.56	IV
25	Good	0.00	20.45	108.00	113.24	-	241.70	542	38.33	IV
26	Best	0.00	134.46	108.00	138.03	-	380.49	1475	149.73	IV
26	Good	0.00	215.21	108.00	139.58	-	462.79	774	69.13	IV
27	Best	908.6	83.76	108.00	151.47	-	1251.82	1488	194.77	III
27	Good	916.3	170.70	108.00	151.74	-	1346.75	359	38.13	III
30	Best	0.00	92.24	108.00	136.83	-	337.08	1337	136.11	III
30	Good	0.00	158.32	108.00	139.00	-	405.33	632	58.50	III
33	Best	882.2	0.59	108.00	150.29	-	1141.09	1490	239.63	III
33	Good	1033	8.74	108.00	151.06	-	1300.79	477	55.13	III
36	Best	0.00	2.78	107.50	112.34	-	222.62	1459	145.75	III
36	Good	0.00	8.08	108.00	117.18	-	233.26	380	33.06	III
37	Best	0.00	5.77	106.00	111.81	3	223.59	1361	179.80	IV
37	Good	0.00	13.06	108.00	113.34	5	239.41	160	15.96	IV
38	Best	0.00	417.55	107.50	141.95	0	667.00	1478	171.45	III
38	Good	0.00	480.40	107.50	142.33	7	737.23	630	64.65	III
39	Best	1198.7	112.99	108.00	143.83	0	1563.51	1494	202.43	III
39	Good	1256.5	263.85	108.00	143.14	0	1771.49	280	26.90	III

8.1.2 General Observations

The GTTS TDVRSP algorithm performed well and consistently for the thirty-nine benchmark problems. It found good solutions within very acceptable processing times and continued finding better solutions with longer execution efforts. The GTTS TDVRSP algorithm displayed its ability to diversify and find good solution space partitions, so the intensification process can find even better solutions within the space.

The GTTS TDVRSP algorithm found *good* solutions, where all demands were filled and relatively small TDD shortcomings, quickly during the search process. Identifying a good solution was a subjective process, which depended on the author's relative satisfaction with the objective function values. Good solutions shown in Tables 8.1, 8.2, and 8.3 reduced both demand shortfall and TDD shortfall within a reasonable search time. The search times varied for each problem instance, based on problem size and density. Good solutions for small size problems were found within 3 minutes 92% of the time. For medium size problems, they were found within 25 minutes 83% of the time and for large problems, they were found within 60 minutes 80% of the time. Figures 8.1, 8.2, and 8.3 are histograms that display solve time frequencies and cumulative solve time percentages for the good solutions posted in Tables 8.1, 8.2, and 8.3.

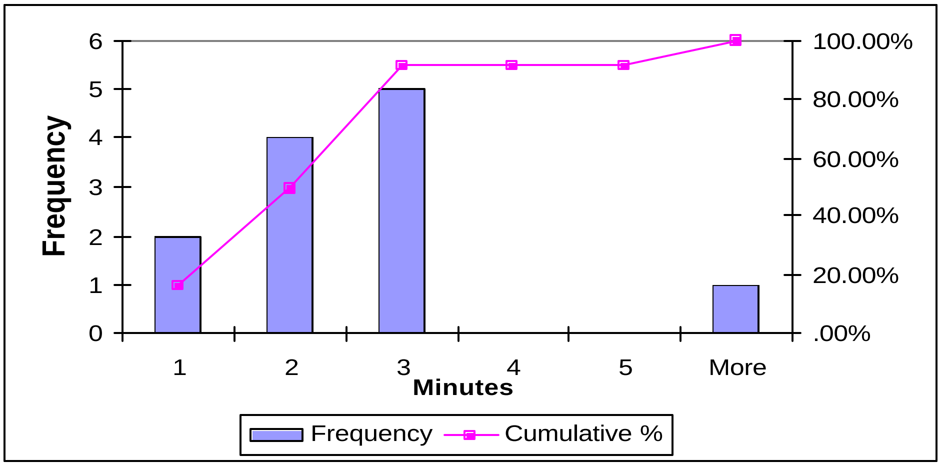

Figure 8.1 Times to Find Good Solutions, Small Size Problems

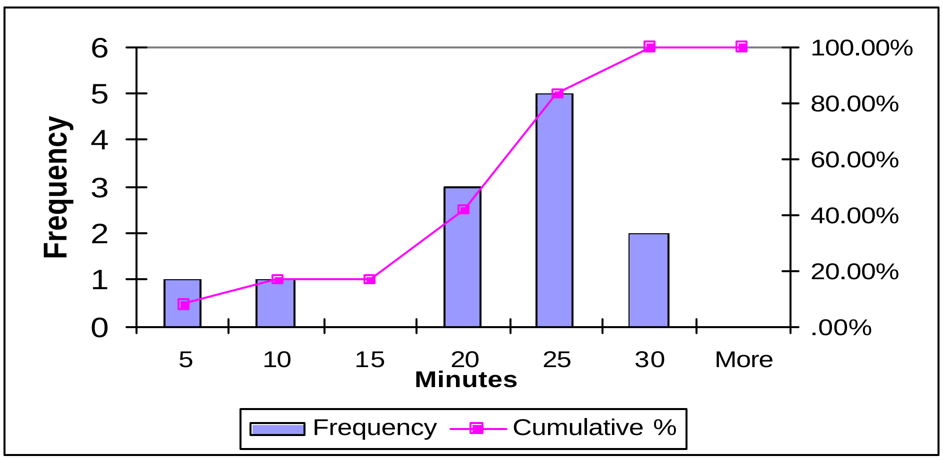

Figure 8.2 Times to Find Good Solutions Medium Size Problems

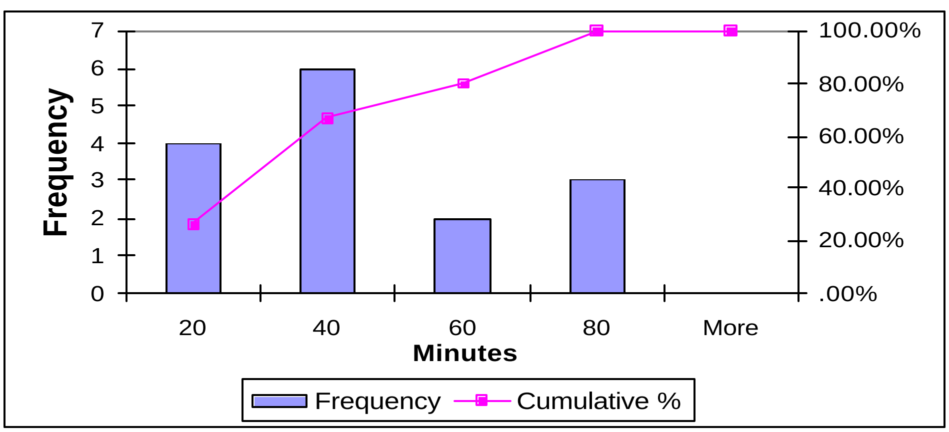

Figure 8.3 Times to Find Good Solutions, Large Size Problems

The solution values were highly dependent on problem density. Higher densities produced larger objective function values in most solutions found for the thirty-nine problems. Figures 8.4, 8.5, and 8.6 are bar charts that display the best solutions' TDD objective function values for problems 1 to 9, 10 to 18, and 19 to 27, respectively. The three bars grouped together represent low, medium, and high delivery densities for the three categories of demand **:** capacity ratio densities.

If the algorithm consistently finds objective function solution values where the solution quality depends on problem delivery density, then the three bars should increase within each group. In this case, seven of the nine groups display increasing bars. Of the two groups not meeting the criteria, one group was only marginally inconsistent.

Additionally, the low, medium and high problem density bars should increase across the groups relative to each other as the demand **:** capacity ratio density increases. For example, when delivery densities are the same, but the demand **:** capacity density ratio increases, there is less chance customers receive goods at an early time. Of the six instances for the low and medium demand **:** capacity groups, the results hold true five times. The instance that did not hold true was only marginally inconsistent. The large demand **:** capacity groups are excluded because there was not enough vehicle capacity to satisfy all demands. Consequently, not all TDD shortfall amounts are included in the objective function, since there is no way to measure delivery lateness if the delivery was never made.

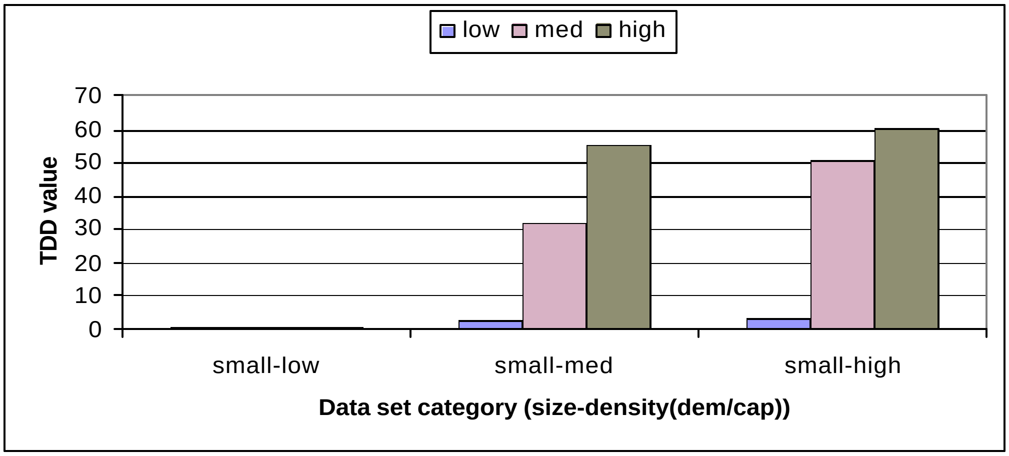

Figure 8.4 TDD Value Relative to Problem Density (Small Problems)

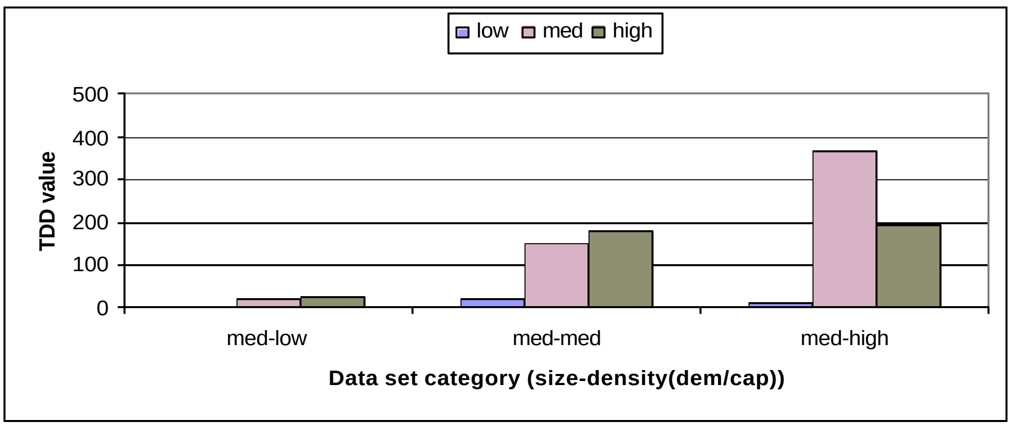

Figure 8.5 TDD Value Relative to Problem Density (Medium Problems)

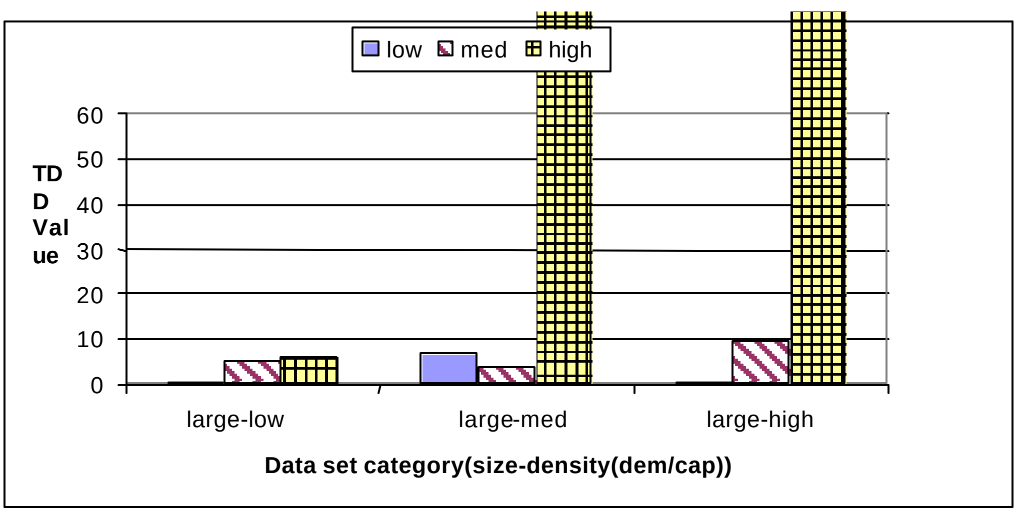

Figure 8.6 TDD Value Relative to Problem Density (Large Problems)

The GTTS TDVRSP algorithm displayed its ability to satisfactorily reduce the multiple objective solution values. The objectives order of importance is demand shortfall, TDD shortfall, fixed costs, and variable costs. The most important objective is to fill customer demands. Tables 8.1, 8.2, and 8.3 reveal the model's ability to reduce demand shortfall. All problem instances with demand **:** capacity ratio less than or equal to one, achieved demand shortfall values equal to zero. Instances with demand **:** capacity ratios greater than one cannot achieve demand shortfalls equal to zero. However, these instances do attempt to reduce demand shortfall to the greatest extent possible.

Along with reducing demand shortfall, the GTTS TDVRSP algorithm methodically reduces the TDD shortfall. In Tables 8.1, 8.2, and 8.3, there are numerous cases where the demand shortfall equals zero for the best and good solution values of a data set instance, but there are differences between TDD shortfall values. This provides evidence of the model's efforts to minimize TDD shortfall once the demand shortfall is, or is nearly, minimized. This effect is more pronounced in the incumbent solution value charts for data sets 11 and 20 in Figures 8.9 and 8.10 of Section 8.3. Once the demand shortfall is minimized, the TDD shortfall gradually decreases.

The GTTS TDVRSP algorithm also reduces the total fixed cost, i.e., the fixed cost associated with the number of vehicles used in the solution. For the thirty-nine benchmark problems, some have demand **:** capacity ratios greater than or equal to one, which are the medium and high density problems. In these cases, it is necessary to use every vehicle to fill customer demands, which is the most important objective. However, nearly one third of the problems have demand **:** capacity ratios less than one and do not

require the use of every vehicle to fill customer demands. Tables 8.1, 8.2, and 8.3 display results where fixed cost values are found below their maximum values. For example, problem 1 had a fixed cost value of 54 in Table 8.1 instead of the maximum value, 62. For problem 1, once the GTTS TDVRSP algorithm minimized the demand shortfall and TDD shortfall, it began searching for solutions that used fewer vehicles, which reduced the fixed cost. Fixed costs are also reduced for problems 10, 13, 19, 22, 25, 29, 31, 36, 37, and 38. Problem 38 is the only case where the fixed cost was reduced and the TDD shortfall was not minimal. This occurred because the network tier, for which the vehicle was removed, had minimized both demand shortfall and TDD shortfall. The overall solution value demand shortfall and TDD shortfall amounts were from other tiers within the instance.

All the instances where fixed cost values were found below their maximum values had low or medium delivery densities. Problems with high delivery densities had difficulty reducing fixed costs because they used all available vehicles to satisfy the more difficult time demands.

For the benchmark problems, variable costs received the lowest priority. Due to the nature of the problem instances, reducing variable costs had marginal impact on the total cost solution values. However, the algorithm does measure the variable costs for each solution, which is pertinent information for planners.

Based on a regression analysis, we suspect the algorithmic run time order of growth has an asymptotic upper bound of $\mathbf{O}(n^2)$. The variable n represents the number of letters in S_n. This measures the normal tabu search iterations' processes.

Figure 8.7 displays run times for 200 normal tabu search iterations for varying problem sizes. The problem sizes consisted of S_{348}, S_{696}, S_{1044}, and S_{1392} and were tested under similar conditions. Problem 5 was used as the baseline. The number of letters increased by increasing the number of vehicles and customers within the same time period.

Each line represents a combination of runs with and without working MOG constraints and orbit tabu lists. The top two lines are the runs that include working MOG constraints. The bottom two lines are the runs without working MOG constraints. Working MOGs were tested because they have the greatest impact on model run time of all the constraints. The orbit tabu lists also impact the run time. Trendlines and equations were found using the EXCEL function *add trendline*.

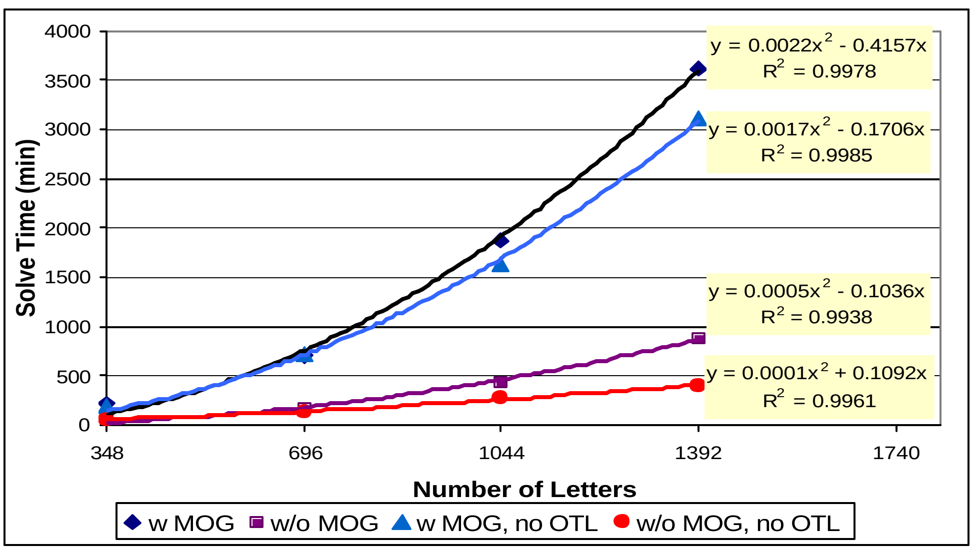

Figure 8.7 Run Time Order of Growth

8.2 Parameter Settings

Several parameters are used to direct the GTTS TDVRSP algorithm. The parameters were first introduced in Section 4.2.1 and later presented in Sections 6.2.1, 6.3.1 and 7.3 for the example problems. The parameters specify a number of thresholds used in the *tabu search strategy manager* and *move neighborhood generator*.

Table 8.4 lists some of the parameters used for testing the benchmark problems. The problems are partitioned into three groups and the table provides the different parameter settings for each data group. Hundreds of runs using various problems were conducted to establish good robust parameter settings. Data groups 1, 2, and 3 are the small, medium and large size problem instances of the benchmark problems, respectively.

Table 8.4 GTTS Parameters for the Data Groups

Parameter	Data group 1	Data group 2	Data group 3
neighborhoodSizeLimit	1000	500	100
iterations	150	250	350
intensificationIterations	50	50	150
maxLoops	20	10	3
moveTabuTenure	3	3	3
worseningMoveTolerance	3	3	5
constantMoveTolerance	5	5	8
intensificationWorseningMoveTolerance	3	3	5
iIntensificationConstantMoveTolerance	5	5	8
superDiversifyRange	200	200	200
superDiversifyTolerance	20	20	30
superDiversifyMoves	6	6	6
objFunctionWeights	{1-3,1,1, .05}	{1-3,1,1, .05}	{1-10,1,1, .05}

Problem size is the primary reason parameters change. For larger problem sizes, there are more potential neighborhood moves. Additionally, the time to evaluate each p^{move} or $p \oplus move$ increases. Consequently, setting parameters that influence the size and frequency of different move neighborhoods becomes more important. For example, evaluating a $p \oplus move$ neighborhood takes up to 40 times longer than a p^{move} neighborhood of equal size. The $p \oplus move$ neighborhood takes longer because the cyclic form of the solution changes for each $p \oplus move$, whereas the cyclic form is maintained for the p^{move} neighborhood. Since the $p \oplus move$ neighborhood is used for diversification, the number of times it is called by the *tabu search strategy manager* is limited for larger problem sizes.

The neighborhood size limit, worsening move tolerance, constant move tolerance, intensification worsening move tolerance, and intensification constant move tolerance parameter settings influence the number of times diversification move neighborhoods are called and the size of each move neighborhood. The smaller problems allow up to a maximum of 1000 moves in a move neighborhood, whereas the large problem only allows up to a maximum of 100 moves. Also, worsening and constant move tolerances, which activate diversification move neighborhoods, are lower for the smaller problem sizes. The tolerances are set at 3 and 5 for the small and medium problems and 5 and 8 for the large problems. These settings were determined as the most robust settings based on numerous tests performed on the problems. They allow sufficient freedom to make worsening and constant moves without straying too far from good regions of the solution space.

The number of normal tabu search iterations and intensification iterations vary for the different problem groups. Problem group 1 parameters direct the GTTS algorithm to conduct 150 normal iterations and 50 intensification iterations. Problem group 2 parameters direct 250 normal iterations and 50 intensification iterations. Problem group 3 parameters are set for 350 normal tabu search iterations and 150 intensification iterations. Normal tabu search iterations increase with problem size to accommodate the increased number of mutually exclusive group neighborhoods for each data group. For the small and medium size problems, the general idea is to allow each tabu search a minimum of five complete cycles through the move neighborhoods during the normal tabu search process and a minimum of one complete cycle for the intensification process. For the large problem, a minimum of three complete cycles and one complete cycle, respectively, are allowed.

The maximum loops and intensification loop parameters prescribe the number of tabu search iterations. The total number of iterations is calculated as

$$\text{Total \# iterations} = maxLoops*(normalTSIter + intenseIter)$$

Given the prescribed parameters, the total number of iterations for data groups 1, 2, and 3 are 4,000, 3,000, and 1,500, respectively.

The objective function weights differ for each benchmark problem within the problem groups. Problems with high delivery densities require a more heavily weighted demand shortfall in the objective function. The demand shortfall is weighted in order to drive the demand shortfall to zero. High delivery densities require a weighted demand

shortfall because very dense problems have high TDD penalties and the solution tends to not deliver cargo rather than delivery it very late. Therefore, penalties for not delivering cargo need to outweigh the penalties for delivering cargo late. The demand shortfall penalty for data group 3 goes up to 10 in order to quickly drive the demand shortfall to zero. This is needed because there are fewer tabu search iterations for data group 3.

8.3 Total Quality Search

Tabu search is based on the premise that intelligent problem solving utilizes adaptive memory and responsive exploration (Glover and Laguna, 1997). Adaptive memory is the use of explicit or attributive memory to guide the search, while responsive exploration is the ability to exploit good solutions while exploring promising regions (Glover and Laguna, 1997). These principles are the underlying philosophies of tabu search that guide tabu search algorithms. However, how does one examine the algorithm's application of these principles? This section provides means to examine the GTTS use of adaptive memory and responsive exploration, which are *total quality search* measures.

Three total quality search measures are presented in this section. The first measure is the *incumbent solution chart,* a display of each incumbent solution's objective function values. This provides a visual account of the search progress through the solution space as measured by the objective function. Figures 8.8, 8.9, and 8.10 are *incumbent solution charts* for three different benchmark problems. These charts are created during the search and are exercised to monitor the search results.

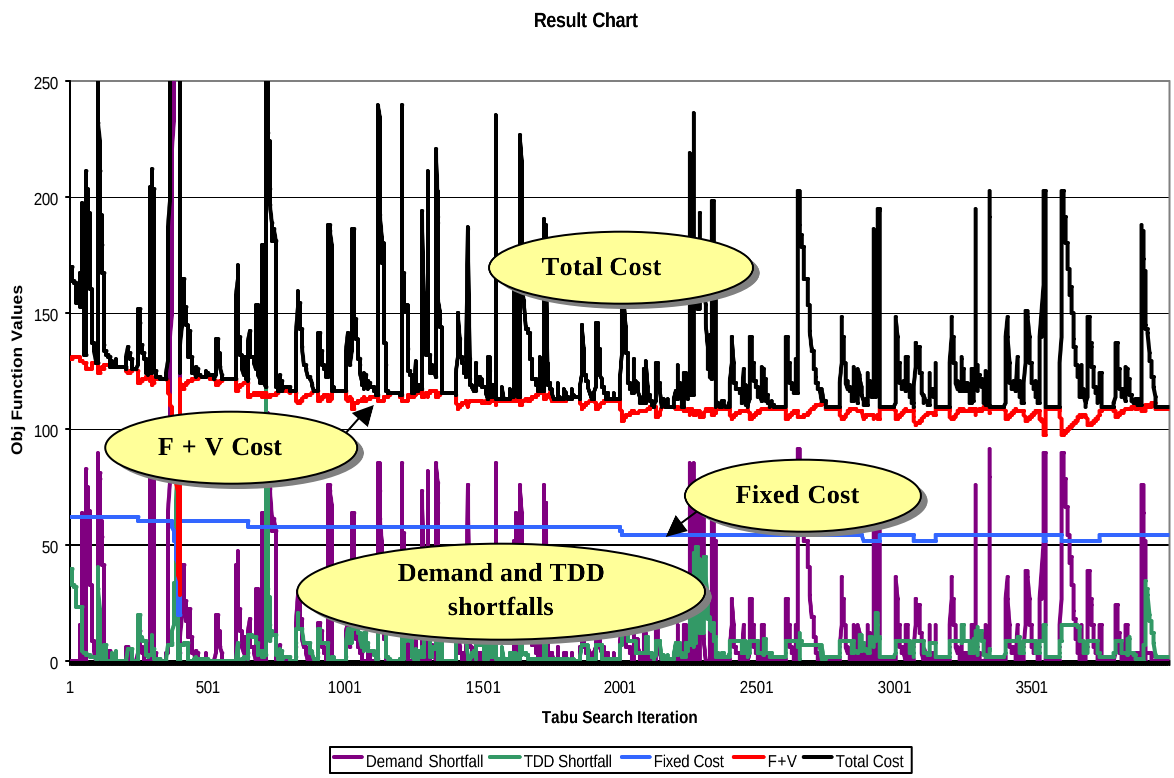

Figure 8.8 Incumbent Solution Chart (Problem 1)

Some interesting characteristics are visually noticeable in Figure 8.8. First, the solution's total cost decreases as the search progresses. Second, a very good solution is found because places where the total cost line and F+V cost line touch indicate the demand shortfall and TDD shortfall are zero. Also, the fixed cost is reduced over time, by deleting vehicles. The chart indicates the algorithm's ability to traverse the solution space with no evidence of cycling.

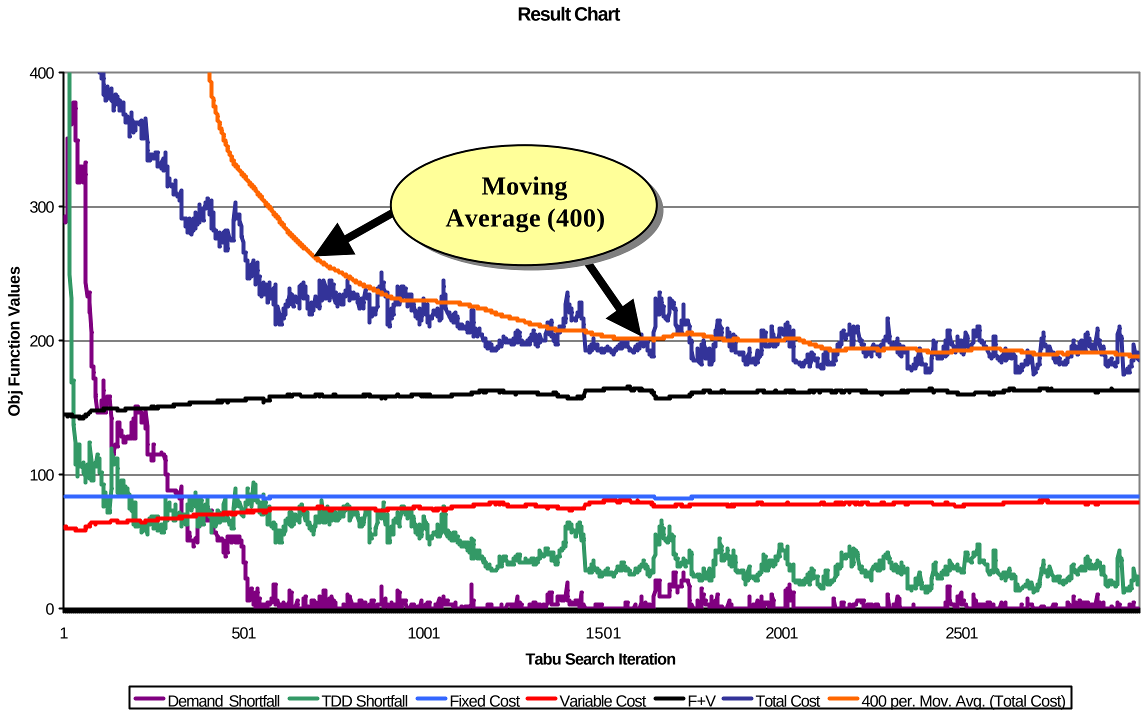

Figure 8.9 Incumbent Solution Chart (Problem16)

Figure 8.9 is the incumbent solution chart for the first 3,000 iterations of problem 16. The top jagged line is the total cost line and the line that cuts through it is a 400 iteration moving average of the total costs. The interesting feature of this chart is the moving average trend. The moving average depicts a downward trend as the iterations progress. This indicates the algorithm is finding improved solutions over time as it intensifies and diversifies across the solution space.

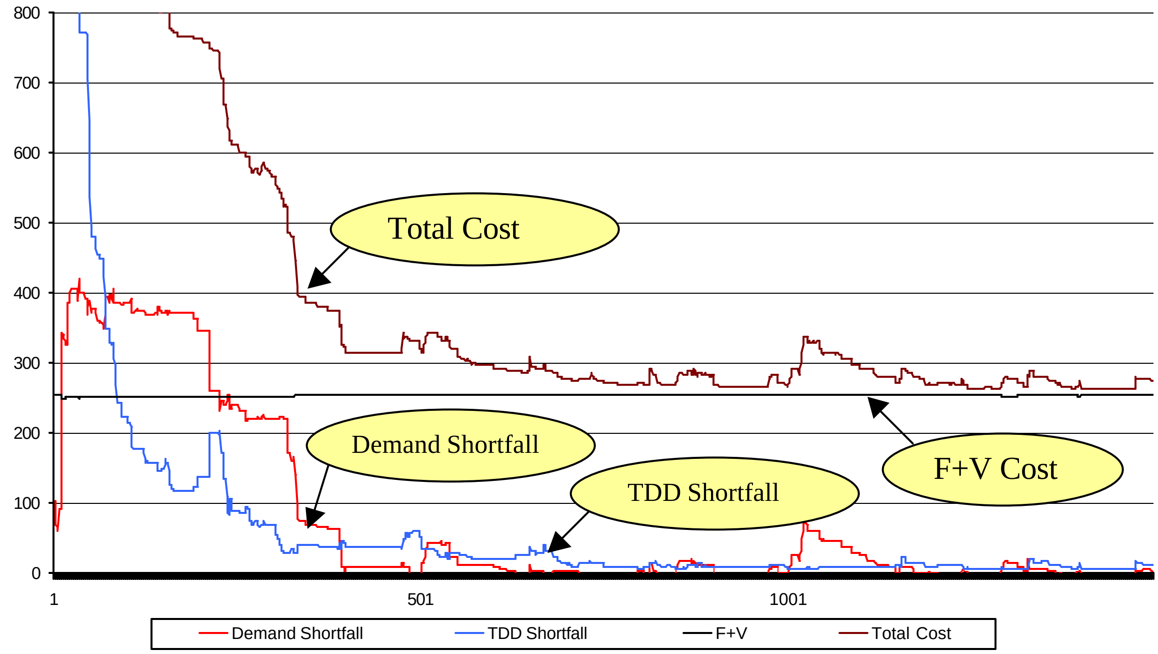

Figure 8.10 Incumbent Solution Chart (Problem 20)

Figure 8.10 is the incumbent solution chart for the first 1,500 iterations of problem 20. Once again, the search indicates its ability to improve over time, while diversifying and intensifying.

The second TQS measure is the *responsive exploration chart*. This chart displays iterations spent exploiting good solutions and exploring other promising regions. It also displays the number of iterations spent in different solution space partitions and the total cost minimum values for each partition. The solution space partitions, for these problems, are conjugacy classes and are represented by bars. The total cost minimum values for each conjugacy class are the diamonds connected by lines. This chart measures two different search characteristics: the number of different conjugacy classes explored during the search and the number of iterations spent intensifying the search around good solutions. Figures 8.11 and 8.12 are two examples of responsive *exploration charts* for the GTTS of problems 5 and 14.

In Figure 8.11, each bar represents a conjugacy class searched during the GTTS of problem 5. There were 47 different conjugacy classes searched over a span of 4,000 iterations. Some conjugacy classes had less than 40 intensification iterations while others had greater than 100 iterations. The diamonds represent the minimum values for each conjugacy class. Note the number of iterations spent around good solutions was high and the number of iterations spent around bad solutions was low.

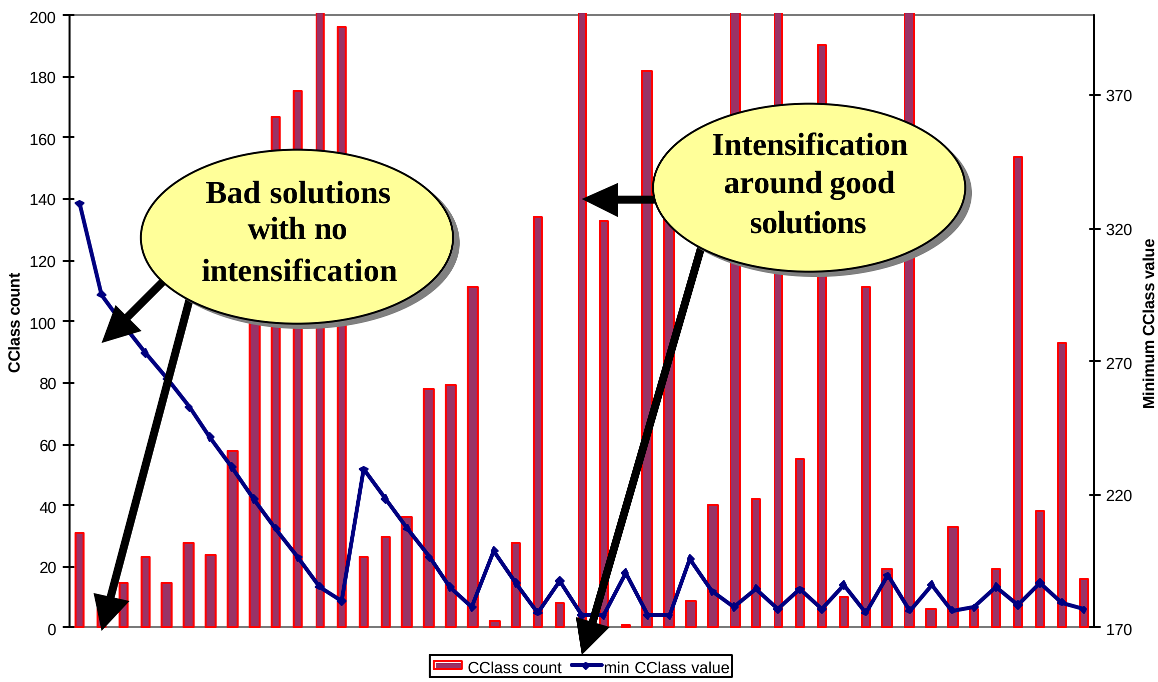

Figure 8.11 Responsive Exploration Chart (Problem 5)

Figure 8.12 is a responsive exploration chart for problem 14. There were 109 different conjugacy classes searched during the 3,000 GTTS iterations. Here again, conjugacy classes with good total cost values had the greatest number of iterations dedicated to intensification efforts.

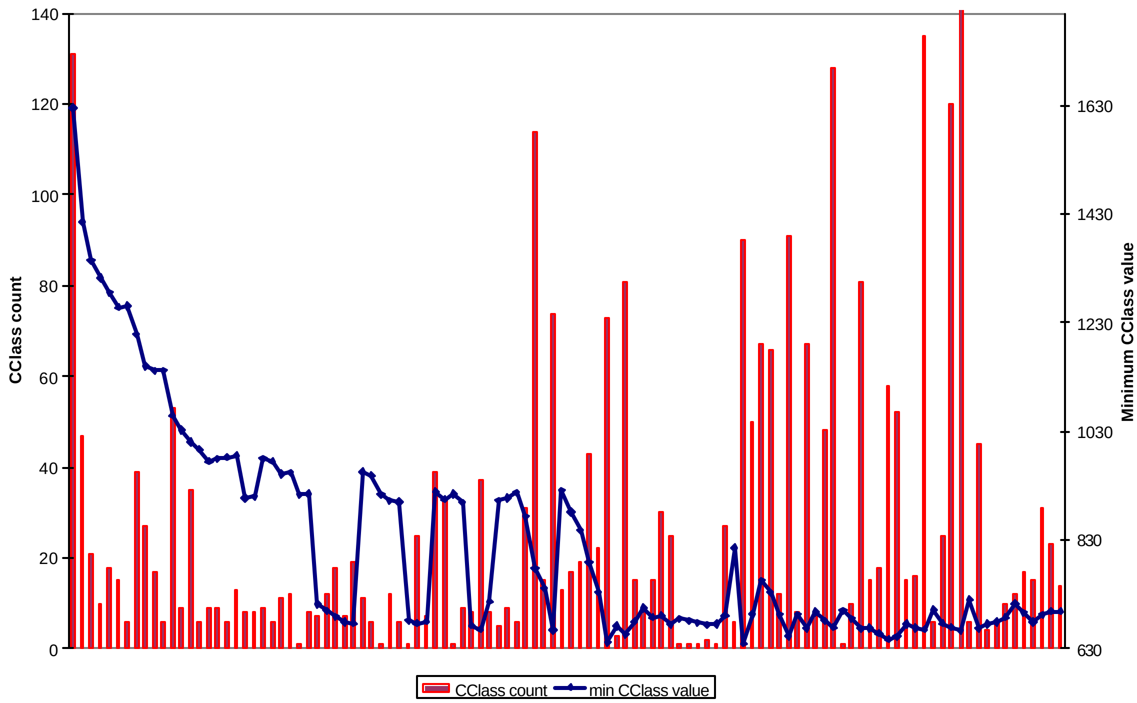

Figure 8.12 Responsive Exploration Chart (Problem 14)

The third TQS measure is the *redundant solution evaluation statistic*. This statistic measures the number of times any solution was reevaluated. The purpose of this statistic is to determine how efficiently the search traverses the solution space.

To determine the *redundant solution evaluation statistic*, tests were performed with a number of benchmark problems. The concept of the test is to run a number of GTTS iterations within the same solution space partition, such that the likelihood of redundant solution evaluations is high. Therefore, the GTTS parameters were set to iterate in the same conjugacy class and cyclic form structure throughout the test. The

search was confined to searching only orbital planes within the same cyclic form structure.

The GTTS algorithm is similar to the one presented in Section 3.7, except no diversification moves are conducted in step 4. The search simply uses group neighborhoods and inter-orbital plane swap move neighborhoods.

The data collected includes all the p^{move} solutions for each iteration. There were approximately 120 p^{move} solutions for each iteration. The data were screened to determine if the same p^{move} was reevaluated in different iterations. If an evaluated p^{move} existed in different iterations, then it was counted as a redundant solution evaluation. The efficiency rating equals $(|p^{move}| - \#redundantSolutions) / |p^{move}|$.

The results were very good with greater than 99.9% efficiency. The only redundancies occurred when inter-orbital plane swap move neighborhoods targeted solutions within an orbit that was already evaluated. Because the orbit was tabu, the p^{move} was not selected as the incumbent, and further redundancies were avoided.

For example using data set 5, out of the 1000 GTTS iterations and 123,062 p^{move} neighbors, there were 0 redundant solution evaluations, indicating a 100% efficiency. Figure 8.13 is a display of the incumbent solution chart for data set 5. Notice that in the last 400 iterations, the search was likely located in a local optimum region, where improvement was marginal. Where typical tabu search algorithms might reevaluate solutions in this space, the GTTS did not. In fact, the GTTS continued to find better solutions in the partition. For instance, iterations 690, 747, 831, and 920 found solutions with diminishing total cost values 281.43, 276.88, 275.50, and 269.70, respectively.

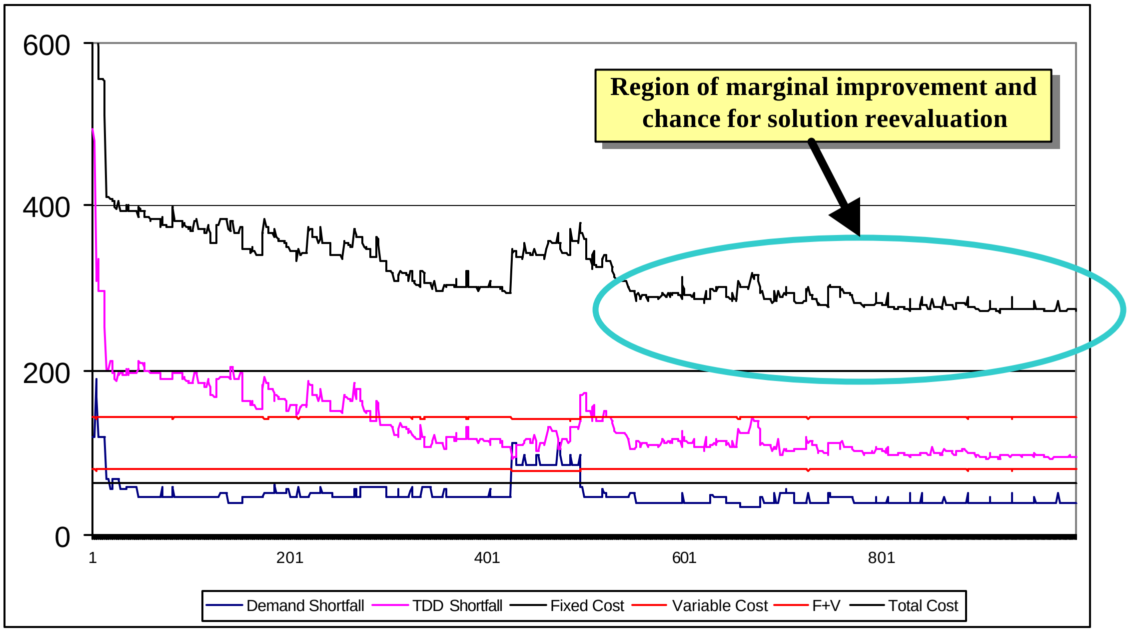

Figure 8.13 GTTS Iterations for Redundant Solution Evaluation Test (Problem 5)

The three total quality search metrics presented in this section were *the incumbent solution chart*, *responsive exploration chart*, and *redundant solution evaluation statistic*. These TQS measures provide a means to evaluate how well tabu search implements the principles of adaptive memory and responsive exploration. Given the results, it is apparent that the GTTS TDVRSP has a quality algorithm that exploits the principles of adaptive memory and responsive exploration.

IX. Areas For Further Research and Summary

This research relating group theory, tabu search, and vehicle routing and scheduling provides interesting and practical approaches to solving large complicated combinatorial problems. Significant insights into realizable approaches for incorporating group theory with tabu search have been gained. A reasonable requirements base and hierarchy for solving multiple theater distribution vehicle routing and scheduling problem instances has been established.

As with many research efforts, this research opens the door for more discoveries in the fields of group theoretic tabu search and theater distribution vehicle routing and scheduling problems. Section 9.1 discusses areas for further research. Part of this section presents improvements to the GTTS TDVRSP algorithm that could enhance performance and functional utilization. Section 9.2 highlights the major contributions of this research effort. Section 9.3 provides a summary.

9.1 Areas For Further Research

There are numerous research areas that could enhance the field of group theoretic tabu search and theater distribution vehicle routing and scheduling problems. They include researching optimal group action generators for move neighborhoods, methods that characterize solution space partitions for exploiting intensification efforts, adding the TDVRSP multi-commodity dimension, enhancing the initial starting solution to consider time requirements, creating a vehicle trip and customer service letter preprocessor, developing graphical user interfaces for model input and output, enhancing the GTTS

TDVRSP algorithm to account for more specific functional requirements, and adopting target analysis as a method to determine which solution space partitions to intensify.

Mutually exclusive groups that generate orbits determine the primary move neighborhoods in the GTTS. They are an important facet of the search efficiency and effectiveness. As discussed in Section 3.4, groups can be developed in numerous ways. Consequently, their composition and generation method becomes an important decision in GTTS. All groups generated in this research were comprised of an exhaustive composition of letter combinations, which was an effective method for the TDVRSP problem. However, are there more efficient groups that can be used within a GTTS? More research should be conducted that determines group composition within a GTTS. Different questions that arise are: Number of letters per group? The group element cyclic form structure? Number of elements per group? Group element generators? Types of groups for different types of combinatorial problems? Research in this area could enhance move neighborhood efficiency of a GTTS.

One of the contributions of this research effort was the characterization of a solution space partition hierarchy. The solution space was partitioned by conjugacy class, cyclic form structure, orbital planes, and orbits. When good solutions were found, the GTTS intensified its search efforts in specific solution space partitions. Sometimes the intensification process found much better solutions, while at other times, the search stalled. Further research should be performed on how to best utilize solution space partitions for finding good solutions. What kind of data should be collected for solution space partitions? How does one best exploit solution space partition data to determine its potential? How does the solution space partition topology compare to normal solution

space topology? What mechanism can determine if the search is stuck in a local optimum or chaotic attractor basin within a solution space partition? Answers to these types of questions could open the door for better GTTS intensification efforts.

The GTTS TDVRSP algorithm solves theater distribution hierarchy instances that include multiple trips, multiple services, and hub dimensions. The dimension not solved was the multi-commodity dimension. The multi-commodity dimension adds a much greater level of difficulty and is an area for further research. However, the multi-commodity dimension was considered when coding the GTTS TDVRSP algorithm, thus providing the architecture necessary for future development. Since the algorithm was coded using object-oriented software, adding multiple commodities is simply a matter of characterizing supply objects by commodity and labeling customer demand by commodity. Supply objects provide in-transit visibility of commodities as they move through the distribution network. The difficulty of adding multiple commodities exists in determining an efficient heuristic for vehicle loading and scheduling. The developer must determine a set of conditions that prescribe when and how each vehicle is loaded at the hub/depot. Since the vehicle loader/scheduler heuristic is providing a vehicle routing plan, the heuristic must determine which commodities to load onto a vehicle that satisfy its customer(s)' demands, should the vehicle receive full or partial loads, wait times for required commodities, and departure schedules. These considerations are in addition to the single commodity heuristic requirements.

Another research topic that could potentially enhance the GTTS TDVRSP algorithm is the development of an initial starting solution that more precisely determines routing and scheduling based on demand and time requirements. Currently, the initial

starting solution primarily routes and schedules vehicles based on demand requirements. Time requirements are minimally considered when prioritizing customers and are based on weighted demand and time requirements. Vehicles are ordered by speed and capacity characteristics. Using ordered customers and vehicles, the initial starting solution heuristic begins assigning vehicles to customers until customer demands are met. Other than time requirements considered when prioritizing customers, customer time requirements are not utilized for initial vehicle schedules. Consequently, initial starting solutions tend to have high TDD shortfall objective function values. Fortunately, the GTTS TDVRSP algorithm performs well in reducing TDD shortfalls as the search progresses.

In the GTTS TDVRSP algorithm pre-tabu search phase, a beneficial enhancement would be a vehicle trip and customer service letter pre-processor. Currently, the input data sets have a defined number of vehicle trip letters and customer service letters. The vehicle trip letters constrain the number of trips a vehicle may make during the model time period. The current data sets have a set number of vehicle trip letters that specify and constrain the demand **:** capacity densities. This is good for the intended purpose, but users may eventually want the program to determine the number of eligible vehicle trips per model time period. A vehicle letter pre-processor could perform this function. Customer service letters provide guidance on the number of vehicle services a customer potentially receives during the model time period. Customer service letters are not constraining to the problem, but do provide placeholders. Determining the number of customer service letters in a problem is a balancing act between providing enough letters as service place holders and providing too many letters that increase the size of the

problem, causing increases in search time. The purpose of a customer service letter pre-processor is to provide that balance.

Graphical User Interfaces (GUIs) are software programs that provide user-friendly means to input program data and parameters, monitor program execution, and interpret program output. Generally, software program functional users desire GUIs as a mechanism to implement programs. The GTTS TDVRSP algorithm does not have a comprehensive set of GUIs to perform the functions mentioned above. Currently, simple EXCEL GUIs exist that graph model output during and post execution as well as provide vehicle and customer output data. Other model output is directed to .txt files. Model input is conducted with .txt files and parameters settings in the Main Java™ program class. It is envisioned that more comprehensive GUIs would enhance model input and output for the functional user. Model GUIs should allow line-by-line data input for customers and vehicles, model parameters, and geographical tools for assessing POD, hub, and customer locations in the theater of operations. GUIs should also provide comprehensive vehicle routing and scheduling plans that include geographical tools for visual assessment.

The final area for further research involves specific algorithmic enhancements to the GTTS TDVRSP algorithm. These enhancements would provide a more granular vehicle routing and scheduling plan. Three specific enhancements include creating methods that determine full or partial vehicle loads, dynamic load and unload durations, and vehicle loads prioritized by TDD requirements. The first enhancement is the creation of a method that determines whether vehicles should leave a depot/hub with a full or partial load. Currently, all vehicles leave the depot/hub with a full load unless the

customer getting serviced requires less, or no supply exists at the depot/hub. This is an appropriate condition for the benchmark problems because many demand **:** capacity densities are greater than or equal to one, thereby requiring full load utilization to fill customer demands. Also, vehicle schedules are constrained more by travel times than by time waiting for supplies to arrive at the depot/hub. However, once conditions are such that vehicle schedules are greatly affected by wait time for supplies, the algorithm should make necessary adjustments. The algorithm must have conditions that determine whether vehicles should wait for full loads or should depart with partial loads. Customer demand and time requirements are likely conditional drivers.

The second GTTS TDVRSP algorithmic enhancement involves creating methods that prescribe dynamic vehicle load and unload times. Currently, the algorithm uses static load and unload times for each vehicle type. Durations are the same for partial or full vehicle loads. The next step is to create dynamic times for each vehicle type based on customer and/or load conditions.

The third GTTS TDVRSP algorithm enhancement is a method that prescribes vehicle load disbursement based on customer TDD requirements. This is a more difficult set of conditions that will potentially cause increased run time. One must determine if the additional run time is worth a potential increase in scheduling efficiency and consequently, better solutions. Currently, the vehicle loader/scheduler heuristic uses one of two options for dispersing vehicle loads. The first method disperses loads to customers prioritized by their position within the trip. The first customer receives as much supply, as it demands, and so on. The second method disperses load equally among all customers in the trip. If a customer does not require equal share, the difference

is dispersed equally to the customers that do demand it. Both methods are based on customer demand. The premise for a new option is to disperse load to customers based on their TDD and demand requirements. Within a vehicle trip, customers competing for supply receive it based on TDD requirement priority. Customers with tighter TDD requirements get all or most supply over customers with looser TDD requirements. As mentioned before, this would not be an easy feature to implement, but it is worth researching.

Target analysis is a method that can be used in the tabu search strategy manager to enhance search efforts. Target analysis provides heuristic search the ability to learn rules that best solve classes of problems (Glover and Laguna, 1997). Rules within the tabu search strategy manager could be established that determine which solution space partition to intensify. Additional links between target analysis and group theory should be further researched.

9.2 Major Contributions

This research effort provided significant advances in the fields of group theoretic tabu search and vehicle routing and scheduling problems. In doing so, application software was developed that efficiently and effectively solved theater distribution vehicle routing and scheduling problems.

Group theoretic tabu search contributions include the formulation of a difficult combinatorial optimization problem in S_n, the definition of move neighborhoods in S_n, and solution space partitions in S_n. In corresponding effort with Wiley (2001), this is the first time a vehicle routing and scheduling problem was formulated as a symmetric group

on n-letters. Using the cyclic form of S_n, each nontrivial cycle contained one vehicle trip letter and one or more customer service letters.

For the first time, move neighborhoods were defined using groups to generate orbits. These orbits were used as a means to efficiently search the solution space. They eliminated cycling, prevented solution reevaluation, and avoided entrapment in local optima.   As a result, the search avoids getting stuck and there is no need for an artificial mechanism to restart the search. The orbits were created to allow exhaustive search of the partition and then were placed on an orbit tabu list to prevent reevaluation.

A unique solution space partition hierarchy was developed using the symmetric group on n-letters. Conjugacy classes, cyclic form structures, orbital planes, and orbits were defined that partition the solution space. Solution space partitions were exploited in the diversification and intensification process. In addition, neighborhoods were constructed to intelligently traverse the partitions and enable a potential search of the entire space. Group move neighborhoods steered the search between different orbits. Swap move neighborhoods traversed the search between different orbital planes. Insert and extraction move neighborhoods moved the search to different conjugacy classes and cyclic form structures.

Orbital planes were defined and used as a primary search mechanism of the GTTS. Orbital planes are orbits partitioned by orbits. They provide a more granular partitioning of the solution space, which permits partial or exhaustive search. The advantage of using orbital planes instead of Colletti's (1999) orbital transversal method is

the ability to re-examine an orbit (orbital plane) without reevaluating solutions within the orbit (orbital plane).

This research provides important contributions to the field of vehicle routing and scheduling problems. First, major functional requirements were collected that define theater distribution vehicle routing and scheduling problems. These requirements expand the dimensions of typical vehicle routing and scheduling problems. The added dimensions are multiple vehicle trips, multiple customer services, and hubs. Given these new dimensions, Carlton's (Carlton, 1995) GVRP hierarchy was expanded to include them for theater distribution vehicle routing and scheduling problems. In addition to these dimensions, a constraint referred to as working MOGs was introduced. Working MOGs constrain the number of vehicles that can service a customer at any given time. This is significant because working MOGs were accounted for by generating first in first out queues at customer locations. Combinatorial optimization methods used to solve vehicle routing and scheduling problems cannot typically account for queuing at a delivery point.

Secondly, a GTTS TDVRSP algorithm was developed that solved TDVRSP hierarchy instances. The algorithm prescribes vehicle routes and schedules that provide economically efficient time definite delivery of goods and services to customers. In doing so, the model maintains total asset visibility (TAV) and in-transit visibility (ITV) of theater assets and cargo. The model is robust enough to account for different modeling objectives, which include both force structure analysis and execution planning. Force structure analysis is the quantifying of assets that satisfactorily meet distribution requirements. Execution planning is the prescription of routes and schedules to meet

customer requirements. The model, due to its flexibility and quickness, can also provide rescheduling and replanning analysis.

It is important to know that many software programs that perform theater distribution modeling are available. However, they are all simulation models and simulations cannot provide either TAV or ITV. Therefore, this model is the first of its kind to offer this functionality. In a recent study funded by HQ USAF (HQ USAF/ILA, 2002), a number of recommendations were made to the Air Force on types of models that could support an automated theater distribution management system. The purpose of the system is to "optimize" the entire theater distribution system, from the number of bases and vehicles, down to the vehicle routes and schedules. They concluded that vehicle routing and scheduling was very difficult to optimize, and development of an optimization approach is considered a high-risk methodology. Therefore, they proposed a low risk method using simulation. Unfortunately, they lose the functional requirements of providing TAV and ITV when using simulation.

This study by HQ USAF validates the importance and magnitude of this research. What was regarded as too difficult was successfully created and developed in this research.

9.3 Summary

Group theoretic tabu search provides an efficient and effective modeling methodology to solve a very difficult combinatorial optimization problem, e.g. the theater distribution vehicle routing and scheduling problem. The GTTS TDVRSP algorithm is a Java™ software program built on the Java™ tabu search architecture created by Harder (2000) and utilized Group and Symmetric Group Java™ classes developed by Wiley

(2001). The GTTS TDVRSP algorithm was tested using thirty-nine benchmark problems to determine its efficiency, effectiveness, flexibility, and robustness. The algorithm performed admirably in all categories and forms of measure. Although the GTTS TDVRSP algorithm was designed to solve generalized problem instances, it can be further developed for more specific applications.

Appendix A: Air Force MTMS TDVRSP Results

Table A.1 Vehicle Tour, Schedule, and Loads

Vehicle	Tour	Arr Time	Start Off/Onload	End Off/Onload	Depart	Delivery
0	Direct del	0	10	10	10	
0	7	11.07	11.07	13.07	13.07	85
0		14.13				
1	Direct del	0	20	20	20	
1	3	21.08	21.08	23.08	23.08	85
1	0	23.3	23.3	25.3	25.3	0
1		26.16				
2	Depot	0	20	24	24	
2	4	24.9	24.9	26.9	26.9	84
2	Depot	27.8	29.8	33.8	33.8	
2	1	34.65	34.65	36.65	36.65	85
2	Depot	37.5	39.5	43.5	43.5	
2	0	44.36	44.36	46.36	46.36	84
2	Depot	47.21				
3	Depot	0	0	4	4	
3	3	5.07	5.07	7.07	7.07	85
3	Depot	8.13	10.13	14.13	14.13	
3	1	14.98	14.98	16.98	16.98	85
3	Depot	17.84	19.84	23.84	23.84	
3	2	24.69	24.69	26.69	26.69	85
3	Depot	27.55	29.55	33.55	33.55	
3	0	34.4	34.4	36.4	36.4	85
3	Depot	37.25				
4	Depot	0	30	34	34	
4	3	35.07	35.07	37.07	37.07	85
4	Depot	38.13	40.13	44.13	44.13	
4	3	45.2	45.2	47.2	47.2	77
4	1	47.41	47.94	49.94	49.94	4
4	Depot	50.8				
5	Depot	0	0	4	4	
5	1	4.85	4.85	6.85	6.85	85
5	Depot	7.71	9.71	13.71	13.71	
5	2	14.56	14.56	16.56	16.56	85
5	Depot	17.42	19.42	23.42	23.42	
5	0	24.27	24.27	26.27	26.27	85
5	Depot	27.12	29.12	33.12	33.12	
5	6	34.02	34.02	36.02	36.02	84
5	Depot	36.91	38.91	42.91	42.91	
5	5	43.81	43.81	45.81	45.81	36
5	1	45.94	45.94	47.94	47.94	49
5	Depot	48.8				

Vehicle	Tour	Arr Time	Start Off/Onload	End Off/Onload	Depart	Delivery
6	Depot	0	0	2	2	
6	3	3.52	3.52	4.52	4.52	12
6	Depot	6.04	7.04	9.04	9.04	
6	2	10.26	10.26	11.26	11.26	12
6	Depot	12.48	14.57	16.57	16.57	
6	0	17.78	18.92	19.92	19.92	12
6	Depot	21.13	24	26	26	
6	5	27.28	27.7	28.7	28.7	12
6	Depot	29.98	33.8	35.8	35.8	
6	2	37.02	37.35	38.35	38.35	12
6	Depot	39.57				
7	Depot	0	0	2	2	
7	4	3.28	3.28	4.28	4.28	12
7	Depot	5.56	6.56	8.56	8.56	
7	2	9.78	9.78	10.78	10.78	12
7	Depot	12	14.13	16.13	16.13	
7	0	17.35	17.92	18.92	18.92	12
7	Depot	20.13	23.42	25.42	25.42	
7	5	26.7	26.7	27.7	27.7	12
7	1	27.89	27.89	28.89	28.89	0
7	Depot	30.11	34.01	36.01	36.01	
7	2	37.22	37.78	38.78	38.78	12
7	Depot	40				
8	Depot	0	2	4	4	
8	2	5.22	6	7	7	12
8	Depot	8.22	9.22	11.22	11.22	
8	4	12.5	12.5	13.5	13.5	12
8	Depot	14.78	15.78	17.78	17.78	
8	0	18.99	19.92	20.92	20.92	12
8	Depot	22.13	25.85	27.85	27.85	
8	1	29.06	29.22	30.22	30.22	12
8	Depot	31.43	35.81	37.81	37.81	
8	2	39.03	39.35	40.35	40.35	12
8	Depot	41.57				
9	Depot	0	2	4	4	
9	4	5.28	5.28	6.28	6.28	12
9	Depot	7.56	8.56	10.56	10.56	
9	3	12.08	12.08	13.08	13.08	12
9	Depot	14.6	15.6	17.6	17.6	
9	2	18.82	18.82	19.82	19.82	12
9	Depot	21.04	23.84	25.84	25.84	
9	5	27.12	27.27	28.27	28.27	12
9	Depot	29.55	33.13	35.13	35.13	
9	2	36.35	36.35	37.35	37.35	12
9	Depot	38.57				

Vehicle	Tour	Arr Time	Start Off/Onload	End Off/Onload	Depart	Delivery
10	Depot	0	4	6	6	
10	6	7.27	7.27	8.27	8.27	12
10	Depot	9.55	10.56	12.56	12.56	
10	0	13.77	14	15	15	12
10	Depot	16.21	17.6	19.6	19.6	
10	2	20.82	20.82	21.82	21.82	12
10	Depot	23.04	26.01	28.01	28.01	
10	1	29.22	30.22	31.22	31.22	12
10	Depot	32.43	36.01	38.01	38.01	
10	2	39.23	39.78	40.78	40.78	12
10	Depot	42				
11	Depot	0	4	6	6	
11	0	7.21	12	13	13	12
11	1	13.05	13.05	14.05	14.05	0
11	Depot	15.26	16.57	18.57	18.57	
11	2	19.79	19.79	20.79	20.79	12
11	Depot	22.01	25.43	27.43	27.43	
11	5	28.7	28.7	29.7	29.7	12
11	Depot	30.98	35.56	37.56	37.56	
11	2	38.78	38.78	39.78	39.78	12
11	1	39.83	39.83	40.83	40.83	0
11	Depot	42.04	43.04	45.04	45.04	
11	7	46.56	46.56	47.56	47.56	11
11	Depot	49.08				
12	Depot	0	4	6	6	
12	0	7.21	13	14	14	12
12	Depot	15.21	16.21	18.21	18.21	
12	1	19.43	19.43	20.43	20.43	12
12	Depot	21.64	25	27	27	
12	1	28.22	28.22	29.22	29.22	12
12	Depot	30.43	35.13	37.13	37.13	
12	2	38.35	38.35	39.35	39.35	12
12	Depot	40.57	42	44	44	
12	7	45.52	45.52	46.52	46.52	12
12	Depot	48.04				
13	Depot	0	4	6	6	
13	3	7.52	7.52	8.52	8.52	12
13	Depot	10.04	11.22	13.22	13.22	
13	2	14.44	14.44	15.44	15.44	12
13	Depot	16.66	17.78	19.78	19.78	
13	0	21	21	22	22	12
13	Depot	23.21	27	29	29	
13	1	30.22	31.22	32.22	32.22	12
13	Depot	33.43	37.14	39.14	39.14	
13	2	40.36	40.36	41.36	41.36	12
13	Depot	42.58				

Vehicle	Tour	Arr Time	Start Off/Onload	End Off/Onload	Depart	Delivery
14	Depot	0	6	8	8	
14	6	9.27	9.27	10.27	10.27	12
14	Depot	11.55	13.22	15.22	15.22	
14	3	16.74	16.74	17.74	17.74	12
14	Depot	19.26	23	25	25	
14	5	26.27	26.27	27.27	27.27	12
14	1	27.46	27.46	28.46	28.46	0
14	Depot	29.68	33.56	35.56	35.56	
14	2	36.78	36.78	37.78	37.78	12
14	Depot	39	40	42	42	
14	7	43.51	43.51	44.51	44.51	12
14	Depot	46.03				
15	Depot	0	6	8	8	
15	6	9.27	9.27	10.27	10.27	12
15	Depot	11.55	13.71	15.71	15.71	
15	0	16.92	16.92	17.92	17.92	12
15	Depot	19.13	20.99	22.99	22.99	
15	5	24.27	24.27	25.27	25.27	12
15	Depot	26.54	27.85	29.85	29.85	
15	5	31.13	31.13	32.13	32.13	12
15	6	32.17	32.17	33.17	33.17	0
15	Depot	34.44	37.81	39.81	39.81	
15	0	41.03	41.03	42.03	42.03	12
15	Depot	43.24				
16	Depot	0	12.56	14.56	14.56	
16	0	15.78	15.78	16.78	16.78	12
16	Depot	17.99	18.99	20.99	20.99	
16	2	22.21	22.21	23.21	23.21	12
16	Depot	24.43	27.43	29.43	29.43	
16	1	30.65	32.22	33.22	33.22	12
16	Depot	34.43	37.57	39.57	39.57	
16	2	40.79	40.79	41.79	41.79	6
16	0	41.88	42.03	43.03	43.03	6
16	Depot	44.24				

Table A.2 Customer Delivery Schedule

Customer	CustLet	Veh	VehLet	Delivery	Arr	Offload	Dep
0	70	11	41	12	7.21	12	13
0	73	12	46	12	7.21	13	14
0	86	10	37	12	13.77	14	15
0	82	16	66	12	15.78	15.78	16.78
0	76	15	62	12	16.92	16.92	17.92
0	80	7	23	12	17.35	17.92	18.92
0	72	6	18	12	17.78	18.92	19.92
0	74	8	28	12	18.99	19.92	20.92
0	79	13	53	12	21	21	22
0	71	5	13	85	24.27	24.27	26.27
0	83	3	8	85	34.4	34.4	36.4
0	88	15	65	12	41.03	41.03	42.03
0	77	16	69	6	41.88	42.03	43.03
0	75	2	4	84	44.36	44.36	46.36
1	90	5	11	85	4.85	4.85	6.85
1	103	3	6	85	14.98	14.98	16.98
1	97	12	47	12	19.43	19.43	20.43
1	92	12	48	12	28.22	28.22	29.22
1	105	8	29	12	29.06	29.22	30.22
1	109	10	39	12	29.22	30.22	31.22
1	100	13	54	12	30.22	31.22	32.22
1	106	16	68	12	30.65	32.22	33.22
1	102	2	3	85	34.65	34.65	36.65
1	94	5	15	49	45.94	45.94	47.94
1	93	4	10	4	47.41	47.94	49.94
2	117	8	26	12	5.22	6	7
2	114	7	22	12	9.78	9.78	10.78
2	113	6	17	12	10.26	10.26	11.26
2	112	13	52	12	14.44	14.44	15.44
2	111	5	12	85	14.56	14.56	16.56
2	119	9	33	12	18.82	18.82	19.82
2	126	11	42	12	19.79	19.79	20.79
2	121	10	38	12	20.82	20.82	21.82
2	127	16	67	12	22.21	22.21	23.21
2	118	3	7	85	24.69	24.69	26.69
2	125	9	35	12	36.35	36.35	37.35
2	129	14	59	12	36.78	36.78	37.78
2	128	6	20	12	37.02	37.35	38.35
2	122	7	25	12	37.22	37.78	38.78
2	123	12	49	12	38.35	38.35	39.35
2	110	11	44	12	38.78	38.78	39.78
2	124	8	30	12	39.03	39.35	40.35
2	115	10	40	12	39.23	39.78	40.78
2	120	13	55	12	40.36	40.36	41.36
2	116	16	69	6	40.79	40.79	41.79

Customer	CustLet	Veh	VehLet	Delivery	Arr	Offload	Dep
3	149	6	16	12	3.52	3.52	4.52
3	131	3	5	85	5.07	5.07	7.07
3	147	13	51	12	7.52	7.52	8.52
3	143	9	32	12	12.08	12.08	13.08
3	144	14	57	12	16.74	16.74	17.74
3	133	1	1	85	21.08	21.08	23.08
3	132	4	9	85	35.07	35.07	37.07
3	134	4	10	77	45.2	45.2	47.2
4	155	7	21	12	3.28	3.28	4.28
4	159	9	31	12	5.28	5.28	6.28
4	151	8	27	12	12.5	12.5	13.5
4	154	2	2	84	24.9	24.9	26.9
5	165	15	63	12	24.27	24.27	25.27
5	167	14	58	12	26.27	26.27	27.27
5	169	7	24	12	26.7	26.7	27.7
5	161	9	34	12	27.12	27.27	28.27
5	163	6	19	12	27.28	27.7	28.7
5	160	11	43	12	28.7	28.7	29.7
5	162	15	64	12	31.13	31.13	32.13
5	164	5	15	36	43.81	43.81	45.81
6	170	10	36	12	7.27	7.27	8.27
6	173	14	56	12	9.27	9.27	10.27
6	172	15	61	12	9.27	9.27	10.27
6	174	5	14	84	34.02	34.02	36.02
7	182	0	0	85	11.07	11.07	13.07
7	183	14	60	12	43.51	43.51	44.51
7	181	12	50	12	45.52	45.52	46.52
7	180	11	45	11	46.56	46.56	47.56

Appendix B: Joint MTMS TDVRSP Results

Table B.1 Vehicle Tour, Schedule, and Loads

Vehicle	Cust/dep	Arr Time	Start Off/Onload	End Off/Onload	Depart	Delivery
0	Direct del	0	10	10	10	
0	3	11.08	11.08	13.08	13.08	85
0		14.16				
1	Direct del	0	20	20	20	
1	3	21.08	21.08	23.08	23.08	85
1		24.16				
2	Depot	0	20.43	24.43	24.43	
2	0	25.28	25.28	27.28	27.28	85
2	Depot	28.14	30.9	34.9	34.9	
2	2	35.76	35.76	37.76	37.76	85
2	Depot	38.62	40.62	44.62	44.62	
2	3	45.68	45.68	47.68	47.68	85
2	Depot	48.75				
3	Depot	0	30	34	34	
3	3	35.07	35.07	37.07	37.07	85
3	Depot	38.13	40.13	44.13	44.13	
3	1	44.98	44.98	46.98	46.98	85
3	Depot	47.84				
4	Depot	0	0	4	4	
4	3	5.07	5.07	7.07	7.07	85
4	Depot	8.13	10.13	14.13	14.13	
4	1	14.98	14.98	16.98	16.98	85
4	Depot	17.84	19.84	23.84	23.84	
4	2	24.69	24.69	26.69	26.69	85
4	Depot	27.55	29.55	33.55	33.55	
4	0	34.4	34.4	36.4	36.4	85
4	Depot	37.25				
5	Depot	0	0	2	2	
5	0	3.21	3.21	4.21	4.21	12
5	Depot	5.43	6.43	8.43	8.43	
5	0	9.64	9.64	10.64	10.64	12
5	Depot	11.85	14	16	16	
5	0	17.22	17.22	18.22	18.22	12
5	Depot	19.43	20.45	22.45	22.45	
5	2	23.67	23.67	24.67	24.67	12
5	Depot	25.89	27.86	29.86	29.86	
5	2	31.08	31.08	32.08	32.08	12
5	Depot	33.3				

Vehicle	Cust/dep	Arr Time	Start Off/Onload	End Off/Onload	Depart	Delivery
6	Depot	0	0	2	2	
6	1	3.22	3.22	4.22	4.22	12
6	1	4.22	4.22	5.22	5.22	0
6	Depot	6.43	7.43	9.43	9.43	
6	2	10.65	10.65	11.65	11.65	12
6	Depot	12.87	16.01	18.01	18.01	
6	3	19.53	19.53	20.53	20.53	12
6	Depot	22.04	24.46	26.46	26.46	
6	0	27.67	27.67	28.67	28.67	12
6	Depot	29.89	33.56	35.56	35.56	
6	1	36.78	36.78	37.78	37.78	12
6	Depot	38.99				
7	Depot	0	0	2	2	
7	0	3.21	4.21	5.21	5.21	12
7	Depot	6.43	7.43	9.43	9.43	
7	1	10.64	10.64	11.64	11.64	12
7	Depot	12.86	14.14	16.14	16.14	
7	2	17.35	17.35	18.35	18.35	12
7	Depot	19.57	22.45	24.45	24.45	
7	2	25.67	25.67	26.67	26.67	12
7	Depot	27.89	28.89	30.89	30.89	
7	0	32.11	32.11	33.11	33.11	12
7	Depot	34.32				
8	Depot	0	2	4	4	
8	1	5.22	5.22	6.22	6.22	12
8	Depot	7.43	8.43	10.43	10.43	
8	1	11.65	12.64	13.64	13.64	12
8	Depot	14.86	16.01	18.01	18.01	
8	0	19.22	19.22	20.22	20.22	12
8	Depot	21.43	23.85	25.85	25.85	
8	2	27.06	27.06	28.06	28.06	12
8	Depot	29.28	31.92	33.92	33.92	
8	2	35.14	35.14	36.14	36.14	12
8	Depot	37.36				
9	Depot	0	2	4	4	
9	1	5.22	6.22	7.22	7.22	12
9	Depot	8.43	9.43	11.43	11.43	
9	2	12.65	12.65	13.65	13.65	12
9	Depot	14.87	16.14	18.14	18.14	
9	2	19.36	19.36	20.36	20.36	12
9	Depot	21.58	24.46	26.46	26.46	
9	1	27.68	27.68	28.68	28.68	12
9	Depot	29.89	33.94	35.94	35.94	
9	0	37.15	37.15	38.15	38.15	12
9	Depot	39.36				

Vehicle	Cust/dep	Arr Time	Start Off/Onload	End Off/Onload	Depart	Delivery
10	Depot	0	2	4	4	
10	0	5.21	5.21	6.21	6.21	12
10	Depot	7.43	8.43	10.43	10.43	
10	1	11.64	11.64	12.64	12.64	12
10	Depot	13.86	16.01	18.01	18.01	
10	1	19.22	19.22	20.22	20.22	12
10	Depot	21.44	24.44	26.44	26.44	
10	2	27.66	27.69	28.69	28.69	12
10	Depot	29.91	34.01	36.01	36.01	
10	0	37.22	38.15	39.15	39.15	12
10	Depot	40.36	41.36	43.36	43.36	
11	Depot	0	12	14	14	
11	0	15.21	15.21	16.21	16.21	12
11	Depot	17.43	18.43	20.43	20.43	
11	1	21.64	21.64	22.64	22.64	12
11	Depot	23.86	25.85	27.85	27.85	
11	2	29.07	29.07	30.07	30.07	12
11	Depot	31.29	34.91	36.91	36.91	
11	2	38.13	38.13	39.13	39.13	12
11	Depot	40.35	41.35	43.35	43.35	
11	0	44.57	44.57	45.57	45.57	12
11	Depot	46.78				
12	Depot	0	12	14	14	
12	2	15.22	15.22	16.22	16.22	12
12	Depot	17.44	18.44	20.44	20.44	
12	0	21.65	21.65	22.65	22.65	12
12	Depot	23.87	26.45	28.45	28.45	
12	2	29.67	29.67	30.67	30.67	12
12	Depot	31.89	35.57	37.57	37.57	
12	0	38.79	39.15	40.15	40.15	12
12	Depot	41.36	42.36	44.36	44.36	
12	3	45.88	45.88	46.88	46.88	12
12	Depot	48.4				
13	Depot	0	12	14	14	
13	2	15.22	15.22	16.22	16.22	12
13	Depot	17.44	18.44	20.44	20.44	
13	2	21.66	21.66	22.66	22.66	12
13	Depot	23.88	26.47	28.47	28.47	
13	3	29.99	29.99	30.99	30.99	12
13	Depot	32.5	36.02	38.02	38.02	
13	1	39.24	39.24	40.24	40.24	12
13	Depot	41.45	43.37	45.37	45.37	
13	1	46.58	46.98	47.98	47.98	12
13	Depot	49.2				

Vehicle	Cust/dep	Arr Time	Start Off/Onload	End Off/Onload	Depart	Delivery
14	Depot	0	14	16	16	
14	1	17.22	17.22	18.22	18.22	12
14	Depot	19.43	20.45	22.45	22.45	
14	0	23.66	23.66	24.66	24.66	12
14	Depot	25.87	26.87	28.87	28.87	
14	2	30.09	30.09	31.09	31.09	12
14	Depot	32.31	35.95	37.95	37.95	
14	0	39.16	40.15	41.15	41.15	12
14	Depot	42.36	44.15	46.15	46.15	
14	0	47.36	47.36	48.36	48.36	8
14	Depot	49.57				
15	Depot	0	14	16	16	
15	1	17.22	18.22	19.22	19.22	12
15	Depot	20.43	22.45	24.45	24.45	
15	2	25.67	26.69	27.69	27.69	12
15	Depot	28.91	29.91	31.91	31.91	
15	3	33.43	33.43	34.43	34.43	12
15	Depot	35.95	36.95	38.95	38.95	
15	0	40.16	41.15	42.15	42.15	12
15	Depot	43.36	44.38	46.38	46.38	
15	3	47.89	47.89	48.89	48.89	11
15	Depot	50.41				
16	Depot	0	0	3	3	
16	1	7.49	7.49	8.49	8.49	28
16	Depot	12.99	14.99	17.99	17.99	
16	6	23.15	23.15	24.15	24.15	28
16	Depot	29.32	31.32	34.32	34.32	
16	7	41.28	41.28	42.28	42.28	28
16	Depot	49.24				
17	Depot	0	0	3	3	
17	0	7.56	7.56	8.56	8.56	28
17	Depot	13.11	15.11	18.11	18.11	
17	6	23.28	23.28	24.28	24.28	28
17	Depot	29.44	31.44	34.44	34.44	
17	2	38.9	38.9	39.9	39.9	28
17	Depot	44.36				
18	Depot	0	0	3	3	
18	7	9.96	9.96	10.96	10.96	28
18	Depot	17.92	20.95	23.95	23.95	
18	6	29.11	29.11	30.11	30.11	28
18	Depot	35.28	37.34	40.34	40.34	
18	7	47.3	47.3	48.3	48.3	28
18	Depot	55.26				
19	Depot	0	0	3	3	
19	4	8.42	8.42	9.42	9.42	28
19	Depot	14.83	17.93	20.93	20.93	
19	6	26.1	26.1	27.1	27.1	28
19	Depot	32.27	34.33	37.33	37.33	
19	6	42.49	42.49	43.49	43.49	28
19	Depot	48.66				

Vehicle	Cust/dep	Arr Time	Start Off/Onload	End Off/Onload	Depart	Delivery
20	Depot	0	3	6	6	
20	2	10.46	10.46	11.46	11.46	28
20	Depot	15.91	17.99	20.99	20.99	
20	6	26.16	26.16	27.16	27.16	28
20	Depot	32.32	34.45	37.45	37.45	
20	5	42.73	42.73	43.73	43.73	28
20	Depot	49.02				
21	Depot	0	6	9	9	
21	4	14.42	14.42	15.42	15.42	28
21	Depot	20.84	22.84	25.84	25.84	
21	1	30.33	30.33	31.33	31.33	28
21	Depot	35.83	37.97	40.97	40.97	
21	4	46.39	46.39	47.39	47.39	28
21	Depot	52.8				
22	Depot	0	7	10	10	
22	4	15.42	15.42	16.42	16.42	28
22	Depot	21.84	23.96	26.96	26.96	
22	1	31.45	31.45	32.45	32.45	28
22	Depot	36.94	39.83	42.83	42.83	
22	4	48.25	48.25	49.25	49.25	28
22	Depot	54.67				
23	Depot	0	7.01	9.01	9.01	
23	7	14.23	14.23	15.23	15.23	16
23	Depot	20.45	21.95	23.95	23.95	
23	5	27.91	27.91	28.91	28.91	16
23	Depot	32.87	34.25	36.25	36.25	
23	4	40.31	40.31	41.31	41.31	16
23	Depot	45.37	46.37	48.37	48.37	
23	4	52.44	53.25	54.25	54.25	16
23	Depot	58.31				
24	Depot	0	3	5	5	
24	0	8.42	8.42	9.42	9.42	16
24	Depot	12.83	13.83	15.83	15.83	
24	7	21.05	21.05	22.05	22.05	16
24	Depot	27.27	28.27	30.27	30.27	
24	0	33.69	33.69	34.69	34.69	16
24	Depot	38.11	40.31	42.31	42.31	
24	0	45.73	45.73	46.73	46.73	16
24	Depot	50.15				
25	Depot	0	3	5	5	
25	7	10.22	10.96	11.96	11.96	16
25	Depot	17.18	18.18	20.18	20.18	
25	6	24.05	24.05	25.05	25.05	16
25	Depot	28.93	30.07	32.07	32.07	
25	0	35.49	35.49	36.49	36.49	16
25	Depot	39.91	40.98	42.98	42.98	
25	3	48.01	48.01	49.01	49.01	16
25	Depot	54.03				

Vehicle	Cust/dep	Arr Time	Start Off/Onload	End Off/Onload	Depart	Delivery
26	Depot	0	3	5	5	
26	5	8.96	8.96	9.96	9.96	16
26	Depot	13.92	14.92	16.92	16.92	
26	4	20.99	20.99	21.99	21.99	16
26	Depot	26.05	27.05	29.05	29.05	
26	1	32.42	32.45	33.45	33.45	16
26	Depot	36.82	37.82	39.82	39.82	
26	4	43.88	43.88	44.88	44.88	16
26	Depot	48.95				
27	Depot	0	8	10	10	
27	1	13.37	13.37	14.37	14.37	16
27	Depot	17.74	18.94	20.94	20.94	
27	2	24.29	24.29	25.29	25.29	16
27	Depot	28.63	29.63	31.63	31.63	
27	6	35.5	35.5	36.5	36.5	16
27	Depot	40.38	42.33	44.33	44.33	
27	7	49.55	49.55	50.55	50.55	14
27	6	51.9	51.9	52.9	52.9	2
27	Depot	56.78				
28	Depot	0	9.01	11.01	11.01	
28	5	14.97	15.96	16.96	16.96	16
28	Depot	20.93	22.01	24.01	24.01	
28	0	27.42	27.42	28.42	28.42	16
28	Depot	31.84	32.93	34.93	34.93	
28	1	38.3	38.61	39.61	39.61	16
28	Depot	42.98	43.98	45.98	45.98	
28	4	50.04	50.04	51.04	51.04	16
28	7	52.4	52.4	53.4	53.4	0
28	Depot	58.62				
29	Depot	0	9.01	11.01	11.01	
29	7	16.23	16.23	17.23	17.23	16
29	Depot	22.45	23.96	25.96	25.96	
29	5	29.92	29.92	30.92	30.92	16
29	Depot	34.88	35.96	37.96	37.96	
29	2	41.3	41.3	42.3	42.3	16
29	Depot	45.64	46.64	48.64	48.64	
29	5	52.6	52.6	53.6	53.6	16
29	Depot	57.57				
30	Depot	0	5	6	6	
30	4	10.07	10.07	11.07	11.07	12
30	Depot	15.13	16.93	17.93	17.93	
30	2	21.27	21.27	22.27	22.27	12
30	Depot	25.62	26.62	27.62	27.62	
30	7	32.84	32.84	33.84	33.84	12
30	Depot	39.06	40.35	41.35	41.35	
30	6	45.23	45.23	46.23	46.23	12
30	Depot	50.1				

Vehicle	Cust/dep	Arr Time	Start Off/Onload	End Off/Onload	Depart	Delivery
31	Depot	0	5	6	6	
31	5	9.96	9.96	10.96	10.96	12
31	Depot	14.93	15.93	16.93	16.93	
31	7	22.15	22.15	23.15	23.15	12
31	Depot	28.37	29.37	30.37	30.37	
31	7	35.59	35.59	36.59	36.59	12
31	Depot	41.81	42.85	43.85	43.85	
31	3	48.87	49.01	50.01	50.01	12
31	Depot	55.03				
32	Depot	0	5	6	6	
32	5	9.96	9.96	10.96	10.96	12
32	Depot	14.93	16.93	17.93	17.93	
32	5	21.89	21.89	22.89	22.89	12
32	Depot	26.85	27.85	28.85	28.85	
32	7	34.07	34.07	35.07	35.07	12
32	Depot	40.29	41.37	42.37	42.37	
32	6	46.24	46.24	47.24	47.24	12
32	Depot	51.12				
33	Depot	0	10	11	11	
33	5	14.96	14.96	15.96	15.96	12
33	Depot	19.93	21	22	22	
33	5	25.96	25.96	26.96	26.96	12
33	Depot	30.92	31.92	32.92	32.92	
33	6	36.8	36.8	37.8	37.8	12
33	Depot	41.67	42.67	43.67	43.67	
33	1	47.04	47.04	48.04	60	12
33	6	60.82	60.82	61.82	61.82	0
33	Depot	65.69				
34	Depot	0	10.01	11.01	11.01	
34	5	14.97	14.97	15.97	15.97	12
34	Depot	19.93	21.19	22.19	22.19	
34	5	26.15	26.15	27.15	27.15	12
34	Depot	31.12	32.12	33.12	33.12	
34	7	38.34	38.34	39.34	39.34	12
34	Depot	44.56	45.56	46.56	46.56	
34	6	50.43	50.43	51.43	51.43	12
34	Depot	55.3				
35	Depot	0	11.01	12.01	12.01	
35	7	17.23	17.23	18.23	18.23	12
35	Depot	23.45	25.02	26.02	26.02	
35	5	29.98	29.98	30.98	30.98	12
35	Depot	34.95	36.26	37.26	37.26	
35	5	41.22	41.22	42.22	42.22	12
35	Depot	46.18	47.18	48.18	48.18	
35	4	52.25	52.25	53.25	53.25	10
35	Depot	57.31				

Vehicle	Cust/dep	Arr Time	Start Off/Onload	End Off/Onload	Depart	Delivery
36	Depot	0	6	7	7	
36	5	12.28	12.28	13.28	13.28	8
36	Depot	18.57	20.94	21.94	21.94	
36	7	28.9	28.9	29.9	29.9	8
36	Depot	36.86	38.29	39.29	39.29	
36	5	44.57	44.57	45.57	45.57	8
36	Depot	50.85				
37	Depot	0	6	7	7	
37	7	12.22	12.22	13.22	13.22	4
37	Depot	18.44	20.19	21.19	21.19	
37	3	26.21	26.21	27.21	27.21	4
37	Depot	32.24	33.24	34.24	34.24	
37	1	37.61	37.61	38.61	38.61	4
37	Depot	41.98	43	44	44	
37	6	47.87	47.87	48.87	48.87	4
37	Depot	52.75				
38	Depot	0	6	7	7	
38	2	10.35	10.35	11.35	11.35	4
38	1	11.6	11.6	12.6	12.6	0
38	Depot	15.97	17.94	18.94	18.94	
38	4	23	23	24	24	4
38	Depot	28.06	29.06	30.06	30.06	
38	2	33.41	33.41	34.41	34.41	4
38	Depot	37.75	39.3	40.3	40.3	
38	2	43.64	43.64	44.64	44.64	4
38	Depot	47.98				
39	Depot	0	12	13	13	
39	7	18.22	18.23	19.23	19.23	4
39	Depot	24.45	25.85	26.85	26.85	
39	5	30.81	30.92	31.92	31.92	4
39	Depot	35.88	37.27	38.27	38.27	
39	4	42.34	42.34	43.34	43.34	4
39	Depot	47.4	48.4	49.4	49.4	
39	3	54.42	54.42	55.42	55.42	0
39	0	57.11	57.11	58.11	58.11	0
39	1	60	60	61	61	4
39	Depot	64.37				
40	Depot	0	12	13	14.13	
40	6	18	18	19	19	4
40	Depot	22.87	24.01	25.01	25.01	
40	5	28.98	28.98	29.98	29.98	4
40	Depot	33.94	34.95	35.95	35.95	
40	3	40.97	40.97	41.97	41.97	4
40	Depot	47	48	49	49	
40	6	52.87	52.87	53.87	53.87	2
40	2	54.92	54.92	55.92	55.92	2
40	Depot	59.27				

Table B.2 Customer Delivery Schedule

Customer	CustLet	Vehicle	VehLet	Delivery amount	Arr	Offload	Dep
0	175	5	11	12	3.21	3.21	4.21
0	177	7	21	12	3.21	4.21	5.21
0	180	10	36	12	5.21	5.21	6.21
0	161	17	69	28	7.56	7.56	8.56
0	168	24	91	16	8.42	8.42	9.42
0	187	5	12	12	9.64	9.64	10.64
0	181	11	41	12	15.21	15.21	16.21
0	182	5	13	12	17.22	17.22	18.22
0	184	8	28	12	19.22	19.22	20.22
0	159	12	47	12	21.65	21.65	22.65
0	173	14	57	12	23.66	23.66	24.66
0	170	2	2	85	25.28	25.28	27.28
0	179	28	108	16	27.42	27.42	28.42
0	164	6	19	12	27.67	27.67	28.67
0	171	7	25	12	32.11	32.11	33.11
0	178	24	93	16	33.69	33.69	34.69
0	174	4	10	85	34.4	34.4	36.4
0	160	25	97	16	35.49	35.49	36.49
0	165	9	35	12	37.15	37.15	38.15
0	176	10	40	12	37.22	38.15	39.15
0	162	12	49	12	38.79	39.15	40.15
0	169	14	59	12	39.16	40.15	41.15
0	166	15	64	12	40.16	41.15	42.15
0	163	11	45	12	44.57	44.57	45.57
0	158	24	94	16	45.73	45.73	46.73
0	172	14	60	8	47.36	47.36	48.36
1	202	6	16	12	3.22	3.22	4.22
1	216	8	26	12	5.22	5.22	6.22
1	217	9	31	12	5.22	6.22	7.22
1	201	16	66	28	7.49	7.49	8.49
1	191	7	22	12	10.64	10.64	11.64
1	194	10	37	12	11.64	11.64	12.64
1	192	8	27	12	11.65	12.64	13.64
1	212	27	103	16	13.37	13.37	14.37
1	189	4	8	85	14.98	14.98	16.98
1	198	14	56	12	17.22	17.22	18.22
1	200	15	61	12	17.22	18.22	19.22
1	195	10	38	12	19.22	19.22	20.22
1	196	11	42	12	21.64	21.64	22.64
1	209	9	34	12	27.68	27.68	28.68
1	206	21	82	28	30.33	30.33	31.33
1	205	22	85	28	31.45	31.45	32.45
1	203	26	101	16	32.42	32.45	33.45
1	204	6	20	12	36.78	36.78	37.78
1	197	37	144	4	37.61	37.61	38.61
1	193	28	109	16	38.3	38.61	39.61
1	207	13	54	12	39.24	39.24	40.24
1	215	3	6	85	44.98	44.98	46.98
1	214	13	55	12	46.58	46.98	47.98
1	211	33	130	12	47.04	47.04	60
1	199	39	153	4	60	60	61

Customer	CustLet	Vehicle	VehLet	Delivery amount	Arr	Offload	Dep
2	221	38	146	4	10.35	10.35	11.35
2	243	20	78	28	10.46	10.46	11.46
2	230	6	17	12	10.65	10.65	11.65
2	232	9	32	12	12.65	12.65	13.65
2	220	12	46	12	15.22	15.22	16.22
2	242	13	51	12	15.22	15.22	16.22
2	222	7	23	12	17.35	17.35	18.35
2	224	9	33	12	19.36	19.36	20.36
2	223	30	116	12	21.27	21.27	22.27
2	238	13	52	12	21.66	21.66	22.66
2	247	5	14	12	23.67	23.67	24.67
2	218	27	104	16	24.29	24.29	25.29
2	219	4	9	85	24.69	24.69	26.69
2	241	7	24	12	25.67	25.67	26.67
2	239	15	62	12	25.67	26.69	27.69
2	227	8	29	12	27.06	27.06	28.06
2	244	10	39	12	27.66	27.69	28.69
2	226	11	43	12	29.07	29.07	30.07
2	228	12	48	12	29.67	29.67	30.67
2	229	14	58	12	30.09	30.09	31.09
2	237	5	15	12	31.08	31.08	32.08
2	233	38	148	4	33.41	33.41	34.41
2	234	8	30	12	35.14	35.14	36.14
2	225	2	3	85	35.76	35.76	37.76
2	236	11	44	12	38.13	38.13	39.13
2	235	17	71	28	38.9	38.9	39.9
2	231	29	113	16	41.3	41.3	42.3
2	240	38	149	4	43.64	43.64	44.64
2	246	40	157	2	54.92	54.92	55.92
3	250	4	7	85	5.07	5.07	7.07
3	251	0	0	85	11.08	11.08	13.08
3	260	6	18	12	19.53	19.53	20.53
3	252	1	1	85	21.08	21.08	23.08
3	261	37	143	4	26.21	26.21	27.21
3	253	13	53	12	29.99	29.99	30.99
3	272	15	63	12	33.43	33.43	34.43
3	249	3	5	85	35.07	35.07	37.07
3	255	40	156	4	40.97	40.97	41.97
3	257	2	4	85	45.68	45.68	47.68
3	265	12	50	12	45.88	45.88	46.88
3	259	15	65	11	47.89	47.89	48.89
3	269	25	98	16	48.01	48.01	49.01
3	254	31	122	12	48.87	49.01	50.01

Customer	CustLet	Vehicle	VehLet	Delivery amount	Arr	Offload	Dep
4	278	19	75	28	8.42	8.42	9.42
4	292	30	115	12	10.07	10.07	11.07
4	285	21	81	28	14.42	14.42	15.42
4	284	22	84	28	15.42	15.42	16.42
4	290	26	100	16	20.99	20.99	21.99
4	289	38	147	4	23	23	24
4	275	23	89	16	40.31	40.31	41.31
4	279	39	152	4	42.34	42.34	43.34
4	288	26	102	16	43.88	43.88	44.88
4	273	21	83	28	46.39	46.39	47.39
4	281	22	86	28	48.25	48.25	49.25
4	274	28	110	16	50.04	50.04	51.04
4	276	35	138	10	52.25	52.25	53.25
4	280	23	90	16	52.44	53.25	54.25
5	310	26	99	16	8.96	8.96	9.96
5	293	31	119	12	9.96	9.96	10.96
5	298	32	123	12	9.96	9.96	10.96
5	300	36	139	8	12.28	12.28	13.28
5	296	33	127	12	14.96	14.96	15.96
5	297	34	131	12	14.97	14.97	15.97
5	309	28	107	16	14.97	15.96	16.96
5	295	32	124	12	21.89	21.89	22.89
5	304	33	128	12	25.96	25.96	26.96
5	305	34	132	12	26.15	26.15	27.15
5	294	23	88	16	27.91	27.91	28.91
5	307	40	155	4	28.98	28.98	29.98
5	299	29	112	16	29.92	29.92	30.92
5	306	35	136	12	29.98	29.98	30.98
5	308	39	151	4	30.81	30.92	31.92
5	301	35	137	12	41.22	41.22	42.22
5	311	20	80	28	42.73	42.73	43.73
5	302	36	141	8	44.57	44.57	45.57
5	312	29	114	16	52.6	52.6	53.6
6	328	40	154	4	18	18	19
6	318	16	67	28	23.15	23.15	24.15
6	319	17	70	28	23.28	23.28	24.28
6	326	25	96	16	24.05	24.05	25.05
6	321	19	76	28	26.1	26.1	27.1
6	331	20	79	28	26.16	26.16	27.16
6	320	18	73	28	29.11	29.11	30.11
6	332	27	105	16	35.5	35.5	36.5
6	330	33	129	12	36.8	36.8	37.8
6	324	19	77	28	42.49	42.49	43.49
6	314	30	118	12	45.23	45.23	46.23
6	317	32	126	12	46.24	46.24	47.24
6	329	37	145	4	47.87	47.87	48.87
6	316	34	134	12	50.43	50.43	51.43
6	315	27	106	2	51.9	51.9	52.9
6	313	40	157	2	52.87	52.87	53.87

Customer	CustLet	Vehicle	VehLet	Delivery amount	Arr	Offload	Dep
7	346	18	72	28	9.96	9.96	10.96
7	333	25	95	16	10.22	10.96	11.96
7	344	37	142	4	12.22	12.22	13.22
7	350	23	87	16	14.23	14.23	15.23
7	342	29	111	16	16.23	16.23	17.23
7	336	35	135	12	17.23	17.23	18.23
7	340	39	150	4	18.22	18.23	19.23
7	335	24	92	16	21.05	21.05	22.05
7	341	31	120	12	22.15	22.15	23.15
7	338	36	140	8	28.9	28.9	29.9
7	339	30	117	12	32.84	32.84	33.84
7	334	32	125	12	34.07	34.07	35.07
7	337	31	121	12	35.59	35.59	36.59
7	343	34	133	12	38.34	38.34	39.34
7	351	16	68	28	41.28	41.28	42.28
7	348	18	74	28	47.3	47.3	48.3
7	349	27	106	14	49.55	49.55	50.55

Table B.3 Joint MTMS TDD Violation Details

Customer ID	Late demand	Length of time	Weighted shortfall
0	12	0.64	0.768
0	10	6.21	6.21
0	12	0.65	0.78
0	12	2.65	3.18
0	85	5.28	44.88
0	7	6.42	4.494
0	81	1.4	11.34
0	16	0.72	1.152
1	8	2.36	1.888
1	16	0.37	0.592
1	9	2.98	2.682
1	12	0.64	0.768
1	2	6.68	1.336
1	12	3.77	4.524
1	4	4.61	1.844
1	16	5.6	8.96
1	10	6.23	6.23
1	12	0.04	0.048
1	4	13	5.2
2	12	0.35	0.42
2	7	2.35	1.645
2	12	1.66	1.992
2	16	2.28	3.648
2	85	3.69	31.365
2	12	3.67	4.404
2	9	4.69	4.221
2	12	0.08	0.096
2	4	2.4	0.96
2	12	4.14	4.968
2	11	5.76	6.336
2	16	0.3	0.48
2	4	2.64	1.056
2	2	13.92	2.784
3	85	2.08	17.68
3	4	6.21	2.484
3	4	9.98	3.992
3	12	1.43	1.716
3	78	4.06	31.668
3	11	0.89	0.979
3	16	1	1.6
3	12	2	2.4

Customer ID	Late demand	Length of time	Weighted shortfall
4	20	0.42	0.84
4	6	17.31	10.386
4	28	1.25	3.5
4	16	3.04	4.864
4	10	5.24	5.24
4	16	6.24	9.984
5	8	1.28	1.024
5	2	3.96	0.792
5	12	2.89	3.468
5	10	6.96	6.96
5	16	0.92	1.472
5	12	0.98	1.176
5	4	1.91	0.764
5	4	12.22	4.888
5	16	7.6	12.16
6	28	0.15	0.42
6	28	0.27	0.756
6	16	1	1.6
6	28	3.1	8.68
6	28	3.15	8.82
6	12	6.11	7.332
6	16	0.5	0.8
6	12	1.79	2.148
6	4	7.49	2.996
6	12	2.22	2.664
6	12	3.24	3.888
6	4	4.87	1.948
6	12	7.43	8.916
6	2	8.9	1.78
6	2	9.87	1.974
7	16	1.22	1.952
7	16	3.23	5.168
7	12	4.23	5.076
7	3	5.23	1.569
7	8	5.9	4.72
7	12	9.83	11.796
7	11	11.07	12.177
7	12	0.586	0.7032
7	12	3.33	3.996
7	10	6.27	6.27
7	28	0.3	0.84
7	14	2.54	3.556
Total (weighted)			**414.39**

Appendix C: MTMS w/hub TDVRSP Results

Table C.1 Vehicle Tour, Schedule, and Loads

Vehicle	Tour	Arr Time	Start Off/Onload	End Off/Onload	Depart	Delivery
0	Direct del	0	10	10	10	
0	3	11.08	11.08	13.08	13.08	85
0		14.16				
1	Direct del	0	30	30	30	
1	3	31.08	31.08	33.08	33.08	85
1		34.16				
2	depot	0	20	24	24	
2	3	25.07	25.07	27.07	27.07	85
2	depot	30	32	36	36	
2	3	37.07	37.07	39.07	39.07	85
2	depot	40.13	42.13	46.13	46.13	
2	3	47.2	47.2	49.2	49.2	85
2	depot	54	56	60	60	
2	3	61.07	61.07	63.07	63.07	85
2	depot	64.13	66.13	70.13	70.13	
2	2	70.99	70.99	72.99	78	82
2	depot	78.86				
3	depot	0	0	4	4	
3	3	5.07	5.07	7.07	7.07	85
3	depot	8.13	10.13	14.13	14.13	
3	3	15.2	15.2	17.2	17.2	85
3	depot	18.26	20.26	24.26	30	
3	3	31.07	31.07	33.07	33.07	85
3	depot	34.13	36.13	40.13	40.13	
3	3	41.2	41.2	43.2	43.2	85
3	depot	44.26	46.26	50.26	54	
3	2	54.86	54.86	56.86	56.86	85
3	depot	57.71	59.71	63.71	63.71	
3	3	64.78	64.78	66.78	66.78	85
3	depot	67.84				
4	depot	0	0	4	4	
4	3	5.07	5.07	7.07	7.07	85
4	depot	8.13	10.13	14.13	14.13	
4	3	15.2	15.2	17.2	17.2	85
4	depot	18.26	20.26	24.26	30	
4	3	31.07	31.07	33.07	33.07	85
4	depot	34.13	36.13	40.13	40.13	
4	3	41.2	41.2	43.2	43.2	85
4	depot	44.26	46.26	50.26	54	
4	3	55.07	55.07	57.07	57.07	85
4	depot	58.13	60.13	64.13	64.13	
4	3	65.2	65.2	67.2	67.2	68
4	2	67.41	67.41	69.41	69.41	17
4	depot	70.27				

Vehicle	Tour	Arr Time	Start Off/Onload	End Off/Onload	Depart	Delivery
5	depot	0	0	2	2	
5	1	3.22	3.22	4.22	4.22	12
5	depot	5.43	6.43	8.43	8.43	
5	3	9.95	9.95	10.95	10.95	12
5	depot	12.47	13.47	15.47	15.47	
5	3	16.98	16.98	17.98	17.98	12
5	depot	19.5	20.5	22.5	22.5	
5	3	24.02	24.02	25.02	25.02	12
5	depot	30	31	33	33	
5	3	34.52	34.52	35.52	35.52	12
5	depot	37.04				
6	depot	0	0	2	2	
6	3	3.52	3.52	4.52	4.52	12
6	depot	6.04	7.04	9.04	9.04	
6	3	10.55	10.55	11.55	11.55	12
6	depot	13.07	14.07	16.07	16.07	
6	3	17.59	17.59	18.59	18.59	12
6	depot	20.11	21.11	23.11	23.11	
6	3	24.63	24.63	25.63	25.63	12
6	depot	30	31	33	33	
6	3	34.52	34.52	35.52	35.52	12
6	depot	37.04				
7	depot	0	0	2	2	
7	3	3.52	3.52	4.52	4.52	12
7	depot	6.04	7.04	9.04	9.04	
7	3	10.55	10.95	11.95	11.95	12
7	depot	13.47	14.47	16.47	16.47	
7	3	17.98	17.98	18.98	18.98	12
7	depot	20.5	22.51	24.51	30	
7	3	31.52	33.07	34.07	34.07	12
7	depot	35.58	36.58	38.58	38.58	
7	3	40.1	40.1	41.1	41.1	12
7	depot	42.62	43.62	45.62	45.62	
8	depot	0	0	2	2	
8	1	3.22	3.22	4.22	4.22	12
8	depot	5.43	6.43	8.43	8.43	
8	3	9.95	9.95	10.95	10.95	12
8	depot	12.47	13.47	15.47	15.47	
8	3	16.98	17.2	18.2	18.2	12
8	depot	19.72	20.72	22.72	22.72	
8	3	24.23	24.23	25.23	25.23	12
8	depot	30	31	33	33	
8	3	34.52	35.07	36.07	36.07	12
8	depot	37.58				

Vehicle	Tour	Arr Time	Start Off/Onload	End Off/Onload	Depart	Delivery
9	depot	0	2	4	4	
9	3	5.52	5.52	6.52	6.52	12
9	depot	8.04	9.04	11.04	11.04	
9	3	12.55	12.55	13.55	13.55	12
9	depot	15.07	16.07	18.07	18.07	
9	3	19.59	19.59	20.59	20.59	12
9	depot	22.11	23.11	25.11	30	
9	3	31.52	33.07	34.07	34.07	12
9	depot	35.58	36.58	38.58	38.58	
9	1	39.8	39.8	40.8	40.8	8
9	depot	42.01				
10	depot	0	2	4	4	
10	3	5.52	7.07	8.07	8.07	12
10	depot	9.58	10.58	12.58	12.58	
10	3	14.1	14.1	15.1	15.1	12
10	depot	16.62	17.62	19.62	19.62	
10	3	21.14	21.14	22.14	22.14	12
10	depot	23.66	24.66	26.66	30	
10	3	31.52	33.08	34.08	34.08	12
10	depot	35.6	36.6	38.6	38.6	
10	3	40.12	40.12	41.12	41.12	12
10	depot	42.63				
11	depot	0	2	4	4	
11	3	5.52	7.07	8.07	8.07	12
11	depot	9.58	10.58	12.58	12.58	
11	3	14.1	14.1	15.1	15.1	12
11	depot	16.62	17.62	19.62	19.62	
11	3	21.14	21.14	22.14	22.14	12
11	depot	23.66	24.66	26.66	30	
11	3	31.52	34.07	35.07	35.07	12
11	depot	36.58	37.58	39.58	39.58	
11	2	40.8	40.8	41.8	41.8	12
11	depot	43.02				
12	depot	0	0	3	3	
12	2	7.46	7.46	8.46	8.46	28
12	depot	12.91	14.91	17.91	17.91	
12	2	22.37	22.37	23.37	23.37	28
12	depot	27.83	29.83	32.83	32.83	
12	2	37.28	37.28	38.28	38.28	28
12	depot	42.74	44.74	47.74	47.74	
12	2	52.2	52.2	53.2	53.2	28
12	depot	57.66	59.66	62.66	62.66	
12	0	67.21	67.21	68.21	68.21	28
12	depot	72.77				

Vehicle	Tour	Arr Time	Start Off/Onload	End Off/Onload	Depart	Delivery
13	depot	0	0	3	3	
13	0	7.56	7.56	8.56	8.56	28
13	depot	13.11	15.11	18.11	18.11	
13	2	22.57	23.03	24.03	24.03	28
13	depot	28.49	30.49	33.49	33.49	
13	2	37.94	38.28	39.28	39.28	28
13	depot	43.74	45.74	48.74	48.74	
13	0	53.3	53.3	54.3	54.3	28
13	depot	58.85	60.85	63.85	63.85	
13	0	68.41	68.41	69.41	69.41	28
13	depot	73.96				
14	depot	0	6	9	9	
14	1	13.49	13.49	14.49	14.49	28
14	depot	18.99	20.99	23.99	23.99	
14	1	28.48	28.48	29.48	29.48	28
14	depot	33.98	35.98	38.98	38.98	
14	1	43.47	43.47	44.47	44.47	28
14	depot	48.96	50.96	53.96	53.96	
14	2	58.42	58.42	59.42	59.42	28
14	depot	63.88	65.88	68.88	68.88	
14	0	73.43	73.43	74.43	74.43	28
14	depot	78.99				
15	depot	0	6	9	9	
15	2	13.46	13.46	14.46	14.46	28
15	depot	18.91	20.91	23.91	23.91	
15	1	28.41	28.41	29.41	29.41	28
15	depot	33.9	35.9	38.9	38.9	
15	1	43.4	43.4	44.4	44.4	28
15	depot	48.89	50.89	53.89	53.89	
15	2	58.35	58.35	59.35	59.35	28
15	depot	63.8	65.8	68.8	68.8	
15	0	73.36	73.36	74.36	74.36	28
15	depot	78.92	80.92	83.92	83.92	
16	depot	0	0	2	2	
16	1	5.37	5.37	6.37	6.37	16
16	depot	9.74	10.74	12.74	12.74	
16	2	16.08	16.08	17.08	17.08	16
16	depot	20.43	21.43	23.43	23.43	
16	2	26.77	26.77	27.77	27.77	16
16	depot	31.11	32.11	34.11	34.11	
16	0	37.53	37.53	38.53	38.53	16
16	depot	41.95	42.95	44.95	44.95	
16	0	48.36	48.36	49.36	49.36	16
16	depot	52.78	53.78	55.78	55.78	
16	2	59.12	59.12	60.12	60.12	16
16	depot	63.46				

Vehicle	Tour	Arr Time	Start Off/Onload	End Off/Onload	Depart	Delivery
17	depot	0	0	2	2	
17	1	5.37	5.37	6.37	6.37	16
17	depot	9.74	10.74	12.74	12.74	
17	1	16.11	16.11	17.11	17.11	16
17	depot	20.48	21.48	23.48	23.48	
17	0	26.9	26.9	27.9	27.9	16
17	depot	31.32	32.32	34.32	34.32	
17	2	37.66	37.66	38.66	38.66	16
17	depot	42	43	45	45	
18	depot	0	8	10	10	
18	0	13.42	13.42	14.42	14.42	16
18	depot	17.83	18.83	20.83	20.83	
18	2	24.18	24.77	25.77	25.77	16
18	depot	29.11	30.11	32.11	32.11	
18	1	35.48	35.48	36.48	36.48	16
18	depot	39.85	40.85	42.85	42.85	
18	2	46.2	46.2	47.2	47.2	16
18	0	47.7	47.7	48.7	48.7	0
18	depot	52.11	53.11	55.11	55.11	
18	0	58.53	58.53	59.53	59.53	16
18	depot	62.95				
19	depot	0	8	10	10	
19	2	13.34	13.34	14.34	14.34	16
19	depot	17.69	18.69	20.69	20.69	
19	2	24.03	24.03	25.03	25.03	16
19	depot	28.37	29.37	31.37	31.37	
19	1	34.74	34.74	35.74	35.74	16
19	depot	39.11	40.11	42.11	42.11	
19	0	45.53	45.53	46.53	46.53	16
19	depot	49.95	50.95	52.95	52.95	
19	2	56.29	56.29	57.29	57.29	16
19	depot	60.63				
20	depot	0	0	1	1	
20	2	4.34	4.34	5.34	5.34	12
20	depot	8.69	9.69	10.69	10.69	
20	1	14.06	14.06	15.06	15.06	12
20	depot	18.43	19.43	20.43	20.43	
20	2	23.77	23.77	24.77	24.77	12
20	depot	28.11				
21	depot	0	0	1	1	
21	2	4.34	4.34	5.34	5.34	12
21	depot	8.69	9.69	10.69	10.69	
21	2	14.03	14.34	15.34	15.34	12
21	depot	18.69				
22	depot	0	0	1	1	
22	1	4.37	4.37	5.37	5.37	12
22	depot	8.74	9.74	10.74	10.74	
22	0	14.16	14.16	15.16	15.16	12
22	depot	18.57	19.57	20.57	20.57	
22	2	23.92	23.92	24.92	24.92	12
22	depot	28.26				

Vehicle	Tour	Arr Time	Start Off/Onload	End Off/Onload	Depart	Delivery
23	depot	0	8	9	9	
23	2	12.34	12.34	13.34	13.34	12
23	depot	16.69	17.69	18.69	18.69	
23	2	22.03	22.03	23.03	23.03	12
23	depot	26.37	27.37	28.37	28.37	
23	1	31.74	31.74	32.74	32.74	12
23	depot	36.11	37.11	38.11	38.11	
23	2	41.45	41.45	42.45	42.45	12
23	depot	45.8	46.8	47.8	47.8	
23	2	51.14	51.14	52.14	52.14	12
23	depot	55.48	56.48	57.48	57.48	
24	depot	0	8	9	9	
24	0	12.42	12.42	13.42	13.42	12
24	depot	16.83	17.83	18.83	18.83	
24	2	22.18	22.18	23.18	23.18	12
24	depot	26.52	27.52	28.52	28.52	
24	2	31.86	31.86	32.86	32.86	12
24	depot	36.2	37.2	38.2	38.2	
24	2	41.55	41.55	42.55	42.55	12
24	depot	45.89				
25	depot	0	8	9	9	
25	0	12.42	12.42	13.42	13.42	12
25	depot	16.83	17.83	18.83	18.83	
25	1	22.2	22.2	23.2	23.2	12
25	depot	26.57	27.57	28.57	28.57	
25	2	31.92	31.92	32.92	32.92	12
25	depot	36.26	37.26	38.26	38.26	
25	2	41.6	41.6	42.6	42.6	12
25	depot	45.95				
26	depot	0	8	9	9	
26	2	13.46	13.46	14.46	14.46	8
26	depot	18.91	19.91	20.91	20.91	
26	2	25.37	25.37	26.37	26.37	8
26	depot	30.83	31.83	32.83	32.83	
26	2	37.28	37.28	38.28	38.28	8
26	depot	42.74				
31	depot	0	12	13	13	
31	2	16.34	16.34	17.34	17.34	4
31	depot	20.69	21.69	22.69	22.69	
31	2	26.03	26.03	27.03	27.03	4
31	depot	30.37				
32	depot	0	12	13	13	
32	2	16.34	16.34	17.34	17.34	4
32	depot	20.69	21.69	22.69	22.69	
32	2	26.03	26.03	27.03	27.03	4
32	depot	30.37				

Vehicle	Tour	Arr Time	Start Off/Onload	End Off/Onload	Depart	Delivery
33	depot	0	9.46	11.46	11.46	
33	7	13.36	24	25	25	28
33	depot	26.9	28.9	30.9	30.9	
33	4	32.19	32.59	33.59	33.59	28
33	depot	34.89	39.28	41.28	41.28	
33	5	42.4	42.4	43.4	43.4	28
33	depot	44.51	53.14	55.14	55.14	
33	4	56.44	56.44	57.44	57.44	28
33	depot	58.73	60.73	62.73	62.73	
33	5	63.84	63.84	64.84	64.84	28
33	depot	65.95	67.95	69.95	69.95	
33	5	71.06	71.06	72.06	72.06	28
33	depot	73.18	75.18	77.18	77.18	
33	6	78.47	78.47	79.47	79.47	28
33	depot	80.77	82.77	84.77	84.77	
33	6	86.06	86.06	87.06	87.06	28
33	depot	88.36				
34	depot	0	14.34	16.34	16.34	
34	5	17.45	18.18	19.18	19.18	28
34	depot	20.29	24.18	26.18	26.18	
34	4	27.47	27.65	28.65	28.65	28
34	depot	29.94	31.94	33.94	33.94	
34	6	35.24	37	38	38	28
34	depot	39.3	41.3	43.3	43.3	
34	4	44.59	44.59	45.59	45.59	28
34	depot	46.89	54.2	56.2	56.2	
34	7	58.1	58.1	59.1	59.1	28
34	depot	61	63	65	65	
34	7	66.89	66.89	67.89	67.89	28
34	depot	69.79	71.79	73.79	73.79	
34	6	75.09	75.09	76.09	76.09	28
34	depot	77.38	79.38	81.38	81.38	
34	7	83.28	83.28	84.28	84.28	28
34	depot	86.18				
35	depot	0	15.34	16.34	16.34	
35	5	17.18	17.18	18.18	18.18	16
35	depot	19.01	20.01	21.01	21.01	
35	5	21.84	21.84	22.84	22.84	16
35	depot	23.68	24.68	25.68	25.68	
35	4	26.65	26.65	27.65	27.65	16
35	depot	28.62	29.62	30.62	30.62	
35	4	31.59	31.59	32.59	32.59	16
35	depot	33.56	34.56	35.56	35.56	
35	7	36.99	36.99	37.99	37.99	16
35	depot	39.41	40.41	41.41	41.41	
35	7	42.84	42.84	43.84	43.84	16
35	depot	45.26	46.26	47.26	47.26	
35	6	48.23	48.23	49.23	49.23	16
35	depot	50.2	57.86	58.86	58.86	
35	6	59.83	59.83	60.83	60.83	16
35	depot	61.8				

Vehicle	Tour	Arr Time	Start Off/Onload	End Off/Onload	Depart	Delivery
36	depot	0	15.46	16.46	16.46	
36	5	17.29	17.29	18.29	18.29	12
36	depot	19.12	20.12	21.12	21.12	
36	5	21.96	21.96	22.96	22.96	12
36	depot	23.79	24.79	25.79	25.79	
36	4	26.76	26.76	27.76	27.76	12
36	depot	28.73	29.9	30.9	30.9	
36	4	31.87	31.87	32.87	32.87	12
36	depot	33.84	34.84	35.84	35.84	
36	7	37.27	37.99	38.99	38.99	12
36	depot	40.41	42.8	43.8	43.8	
36	7	45.23	45.23	46.23	46.23	12
36	depot	47.65	57.86	58.86	58.86	
36	6	59.83	60.83	61.83	61.83	12
36	depot	62.8	63.8	64.8	64.8	
36	6	65.77	65.77	66.77	66.77	12
36	depot	67.74				
37	depot	0	15.46	16.46	16.46	
37	5	17.57	18.29	19.29	19.29	8
37	depot	20.4	21.4	22.4	22.4	
37	7	24.3	25	26	26	8
37	depot	27.9	28.9	29.9	29.9	
37	6	31.19	36	37	37	8
37	depot	38.3	39.3	40.3	40.3	
37	4	41.59	41.59	42.59	42.59	8
37	depot	43.89	44.89	45.89	45.89	
37	5	47	47	48	48	8
37	depot	49.11	57.86	58.86	58.86	
37	7	60.76	60.76	61.76	61.76	8
37	depot	63.65	64.65	65.65	65.65	
37	6	66.95	67.77	68.77	68.77	8
37	depot	70.07	71.07	72.07	72.07	
37	4	73.36	73.36	74.36	74.36	8
37	depot	75.66				
38	depot	0	16.34	17.34	17.34	
38	5	18.18	19.18	20.18	20.18	4
38	depot	21.01	22.01	23.01	23.01	
38	7	24.43	26	27	27	4
38	depot	28.42	29.42	30.42	30.42	
38	7	31.85	31.85	32.85	32.85	4
38	depot	34.27	35.27	36.27	36.27	
38	6	37.24	38	39	39	4
38	depot	39.97	40.97	41.97	41.97	
38	5	42.81	42.81	43.81	43.81	4
38	depot	44.64	45.64	46.64	46.64	
38	7	48.06	48.06	49.06	49.06	4
38	depot	50.49	57.86	58.86	58.86	
38	4	59.83	59.83	60.83	60.83	4
38	depot	61.8	62.8	63.8	63.8	
38	6	64.77	64.77	65.77	65.77	4
38	depot	66.74				

Vehicle	Tour	Arr Time	Start Off/Onload	End Off/Onload	Depart	Delivery
39	depot	0	16.34	17.34	17.34	
39	5	18.18	19.29	20.29	20.29	4
39	depot	21.12	22.12	23.12	23.12	
39	5	23.96	23.96	24.96	24.96	4
39	depot	25.79	26.79	27.79	27.79	
39	4	28.76	28.76	29.76	29.76	4
39	depot	30.73	31.73	32.73	32.73	
39	4	33.71	33.71	34.71	34.71	4
39	depot	35.68	36.68	37.68	37.68	
39	4	38.65	38.65	39.65	39.65	4
39	depot	40.62	43.45	44.45	44.45	
39	7	45.88	46.23	47.23	47.23	4
39	depot	48.65	58.86	59.86	59.86	
39	6	60.83	61.83	62.83	62.83	4
39	depot	63.8	64.8	65.8	65.8	
39	6	66.77	66.77	67.77	67.77	4
39	depot	68.74				
40	depot	0	12.07	14.07	14.07	
40	10	16.31	16.31	17.31	17.31	28
40	depot	19.56	22.21	24.21	24.21	
40	10	26.45	26.45	27.45	27.45	28
40	depot	29.7	37.08	39.08	39.08	
40	11	41.3	41.3	42.3	42.3	28
40	depot	44.53	46.58	48.58	48.58	
40	10	50.82	50.82	51.82	51.82	28
40	depot	54.07	65.08	67.08	67.08	
40	10	69.32	69.32	70.32	70.32	28
40	depot	72.57	74.57	76.57	76.57	
40	12	78.82	78.82	79.82	79.82	28
40	depot	82.06				
41	depot	0	12.55	14.55	14.55	
41	10	16.8	17.31	18.31	18.31	28
41	depot	20.56	23.14	25.14	25.14	
41	10	27.38	27.38	28.38	28.38	28
41	depot	30.63	38.08	40.08	40.08	
41	11	42.3	42.3	43.3	43.3	28
41	depot	45.53	47.53	49.53	49.53	
41	10	51.77	51.77	52.77	52.77	28
41	depot	55.02	67.78	69.78	69.78	
41	12	72.03	72.03	73.03	73.03	28
41	depot	75.27	77.27	79.27	79.27	
41	12	81.52	81.52	82.52	82.52	22
41	depot	84.77				

Vehicle	Tour	Arr Time	Start Off/Onload	End Off/Onload	Depart	Delivery
42	depot	0	7.52	9.52	9.52	
42	11	11.74	11.74	12.74	12.74	28
42	depot	14.96	16.96	18.96	18.96	
42	10	21.21	21.21	22.21	22.21	28
42	depot	24.46	29.07	31.07	31.07	
42	8	31.57	31.57	32.57	32.57	28
42	depot	33.06	39.09	41.09	41.09	
42	11	43.31	43.31	44.31	44.31	28
42	depot	46.53	50.2	52.2	52.2	
42	12	54.44	54.44	55.44	55.44	28
42	depot	57.69	67.78	69.78	69.78	
42	12	72.03	72.03	73.03	73.03	28
42	depot	75.27				
43	depot	0	8.07	10.07	10.07	
43	11	12.29	12.29	13.29	13.29	28
43	depot	15.51	18.2	20.2	20.2	
43	10	22.44	22.44	23.44	23.44	28
43	depot	25.69	34.07	36.07	36.07	
43	11	38.29	38.29	39.29	39.29	28
43	depot	41.51	43.51	45.51	45.51	
43	11	47.73	47.73	48.73	48.73	28
43	depot	50.95	58.07	60.07	60.07	
43	12	62.31	62.31	63.31	63.31	28
43	depot	65.56	67.78	69.78	69.78	
43	9	70.28	70.28	71.28	71.28	28
43	depot	71.77				
44	depot	0	8.07	10.07	10.07	
44	11	12.29	12.74	13.74	13.74	28
44	depot	15.96	18.2	20.2	20.2	
44	11	22.42	22.42	23.42	23.42	28
44	depot	25.64	34.07	36.07	36.07	
44	10	38.31	38.31	39.31	39.31	28
44	depot	41.56	43.56	45.56	45.56	
44	11	47.78	47.78	48.78	48.78	28
44	depot	51	58.07	60.07	60.07	
44	12	62.31	62.31	63.31	63.31	28
44	depot	65.56	67.78	69.78	69.78	
44	9	70.28	70.28	71.28	71.28	19
44	depot	71.77				
45	depot	0	8.07	10.07	10.07	
45	11	12.29	13.18	14.18	14.18	28
45	depot	16.41	19.97	21.97	21.97	
45	11	24.19	24.19	25.19	25.19	28
45	depot	27.41	34.07	36.07	36.07	
45	10	38.31	38.31	39.31	39.31	28
45	depot	41.56	44.2	46.2	46.2	
45	11	48.42	48.42	49.42	49.42	28
45	depot	51.64	58.07	60.07	60.07	
45	12	62.31	62.31	63.31	63.31	28
45	depot	65.56	69.79	71.79	71.79	
45	10	74.04	74.04	75.04	75.04	24
45	depot	77.28				

Vehicle	Tour	Arr Time	Start Off/Onload	End Off/Onload	Depart	Delivery
46	depot	0	14.08	15.08	15.08	
46	10	16.77	16.77	17.77	17.77	16
46	depot	19.45	21.97	22.97	22.97	
46	10	24.66	24.66	25.66	25.66	16
46	depot	27.34	34.07	35.07	35.07	
46	10	36.75	36.75	37.75	37.75	16
46	depot	39.44	40.44	41.44	41.44	
46	12	43.12	43.12	44.12	44.12	16
46	depot	45.81	47.21	48.21	48.21	
46	12	49.9	49.9	50.9	50.9	16
46	depot	52.58	64.07	65.07	65.07	
46	12	66.75	66.75	67.75	67.75	16
46	depot	69.44				
47	depot	0	14.08	15.08	15.08	
47	10	16.77	16.77	17.77	17.77	16
47	depot	19.45	22.21	23.21	23.21	
47	8	23.58	23.58	24.58	24.58	16
47	depot	24.95	35.07	36.07	36.07	
47	12	37.76	37.76	38.76	38.76	16
47	depot	40.44	41.44	42.44	42.44	
47	12	44.13	44.13	45.13	45.13	16
47	depot	46.81	47.81	48.81	48.81	
47	8	49.18	49.18	50.18	50.18	11
47	depot	50.56	51.56	52.56	52.56	
47	11	54.22	54.22	55.22	55.22	16
47	depot	56.89				
48	depot	0	9.52	10.52	10.52	
48	11	12.18	12.18	13.18	13.18	16
48	depot	14.85	15.85	16.85	16.85	
48	8	17.22	17.22	18.22	18.22	16
48	depot	18.6	21.42	22.42	22.42	
48	11	24.08	24.08	25.08	25.08	16
48	depot	26.75	36.07	37.07	37.07	
48	12	38.76	38.76	39.76	39.76	16
48	depot	41.44	42.44	43.44	43.44	
48	12	45.13	45.13	46.13	46.13	16
48	depot	47.81	50.2	51.2	51.2	
48	12	52.88	52.88	53.88	53.88	16
48	depot	55.57				
49	depot	0	10.07	11.07	11.07	
49	11	12.73	13.29	14.29	14.29	16
49	depot	15.95	16.95	17.95	17.95	
49	11	19.62	19.62	20.62	20.62	16
49	depot	22.29	26.02	27.02	27.02	
49	11	28.69	28.69	29.69	29.69	16
49	depot	31.35	36.07	37.07	37.07	
49	12	38.76	38.76	39.76	39.76	16
49	depot	41.44	42.44	43.44	43.44	
49	12	45.13	45.13	46.13	46.13	16
49	depot	47.81				

Vehicle	Tour	Arr Time	Start Off/Onload	End Off/Onload	Depart	Delivery
50	depot	0	18.41	19.41	19.41	
50	11	21.08	21.08	22.08	22.08	12
50	depot	23.74	26.23	27.23	27.23	
50	10	28.92	28.92	29.92	29.92	12
50	depot	31.6	36.07	37.07	37.07	
50	12	38.76	38.76	39.76	39.76	12
50	depot	41.44	42.44	43.44	43.44	
50	12	45.13	45.13	46.13	46.13	12
50	depot	47.81	50.2	51.2	51.2	
50	9	51.57	51.57	52.57	52.57	12
50	depot	52.94	64.07	65.07	65.07	
50	11	66.73	66.73	67.73	67.73	10
50	10	67.98	67.98	68.98	68.98	2
50	depot	70.67				
51	depot	0	18.96	19.96	19.96	
51	11	21.63	21.63	22.63	22.63	12
51	depot	24.3	26.63	27.63	27.63	
51	10	29.31	29.31	30.31	30.31	12
51	depot	32	36.08	37.08	37.08	
51	12	38.76	39.76	40.76	40.76	12
51	depot	42.44	43.44	44.44	44.44	
51	12	46.13	46.13	47.13	47.13	12
51	depot	48.81	50.2	51.2	51.2	
51	10	52.88	52.88	53.88	53.88	12
51	depot	55.57				
52	depot	0	10.07	11.07	11.07	
52	11	12.73	13.74	14.74	14.74	12
52	depot	16.41	17.41	18.41	18.41	
52	10	20.09	20.09	21.09	21.09	12
52	depot	22.78	28.07	29.07	29.07	
52	10	30.75	30.75	31.75	31.75	12
52	depot	33.44	38.08	39.08	39.08	
52	12	40.77	40.77	41.77	41.77	12
52	depot	43.45	45.46	46.46	46.46	
52	12	48.14	48.14	49.14	49.14	12
52	depot	50.83				
53	depot	0	10.07	11.07	11.07	
53	11	12.73	14.18	15.18	15.18	12
53	8	16.53	16.53	17.53	17.53	0
53	depot	17.9	20.2	21.2	21.2	
53	10	22.88	22.88	23.88	23.88	12
53	depot	25.57	28.07	29.07	29.07	
53	10	30.75	30.75	31.75	31.75	12
53	depot	33.44	39.09	40.09	40.09	
53	12	41.77	41.77	42.77	42.77	12
53	depot	44.46	46.46	47.46	47.46	
53	12	49.15	49.15	50.15	50.15	12
53	depot	51.84				

Vehicle	Tour	Arr Time	Start Off/Onload	End Off/Onload	Depart	Delivery
54	depot	0	10.52	11.52	11.52	
54	9	12.02	12.02	13.02	13.02	8
54	depot	13.51	14.51	15.51	15.51	
54	10	17.76	17.77	18.77	18.77	8
54	depot	21.01	22.42	23.42	23.42	
54	12	25.67	25.67	26.67	26.67	8
54	depot	28.92	37.08	38.08	38.08	
54	12	40.32	40.32	41.32	41.32	8
54	depot	43.57	45.52	46.52	46.52	
54	10	48.76	48.76	49.76	49.76	8
54	depot	52.01				
55	depot	0	11.07	12.07	12.07	
55	11	14.29	14.74	15.74	15.74	8
55	depot	17.96	20.2	21.2	21.2	
55	10	23.45	23.45	24.45	24.45	8
55	depot	26.69	28.07	29.07	29.07	
55	9	29.56	29.56	30.56	30.56	8
55	depot	31.06	37.08	38.08	38.08	
55	12	40.32	40.32	41.32	41.32	8
55	depot	43.57	45.57	46.57	46.57	
55	12	48.81	48.81	49.81	49.81	8
55	depot	52.06				
56	depot	0	11.07	12.07	12.07	
56	11	14.29	15.18	16.18	16.18	8
56	depot	18.41	21.2	22.2	22.2	
56	12	24.45	24.45	25.45	25.45	8
56	depot	27.7	37.08	38.08	38.08	
56	10	40.32	40.32	41.32	41.32	8
56	depot	43.57	46.21	47.21	47.21	
56	10	49.45	49.45	50.45	50.45	8
56	depot	52.7	64.07	65.07	65.07	
56	12	67.31	67.31	68.31	68.31	8
56	depot	70.56				
57	depot	0	11.07	12.07	12.07	
57	11	14.29	15.29	16.29	16.29	8
57	depot	18.51	21.2	22.2	22.2	
57	10	24.45	24.45	25.45	25.45	8
57	depot	27.7	38.08	39.08	39.08	
57	12	41.33	41.33	42.33	42.33	8
57	depot	44.58	46.53	47.53	47.53	
57	12	49.77	49.77	50.77	50.77	8
57	depot	53.02	64.07	65.07	65.07	
57	10	67.31	67.31	68.31	68.31	8
57	depot	70.56				
58	depot	0	11.52	12.52	12.52	
58	11	14.19	14.29	15.29	15.29	4
58	depot	16.95	19.41	20.41	20.41	
58	10	22.1	22.1	23.1	23.1	4
58	depot	24.78	28.07	29.07	29.07	
58	10	30.75	30.75	31.75	31.75	4
58	depot	33.44	39.09	40.09	40.09	
58	12	41.77	41.77	42.77	42.77	4

Vehicle	Tour	Arr Time	Start Off/Onload	End Off/Onload	Depart	Delivery
59	depot	0	12.07	13.07	13.07	
59	11	14.73	15.74	16.74	16.74	4
59	depot	18.41	20.41	21.41	21.41	
59	12	23.1	23.1	24.1	24.1	4
59	depot	25.79	29.07	30.07	30.07	
59	10	31.76	31.76	32.76	32.76	4
59	depot	34.44	40.09	41.09	41.09	
59	8	41.46	41.46	42.46	42.46	4
59	depot	42.83	44.45	45.45	45.45	
59	12	47.13	47.13	48.13	48.13	4
59	depot	49.82				
60	depot	0	18.31	19.31	19.31	
60	13	19.55	19.55	20.55	20.55	16
60	depot	20.79	23.21	24.21	24.21	
60	16	24.71	24.71	25.71	25.71	16
60	depot	26.21	28.45	29.45	29.45	
60	16	29.95	29.95	30.95	30.95	15
60	depot	31.46	38.75	39.75	39.75	
60	17	40.25	40.25	41.25	41.25	16
60	depot	41.75	52.82	53.82	53.82	
60	15	53.91	53.91	54.91	54.91	16
60	depot	54.99	76.04	77.04	77.04	
60	18	77.56	77.56	78.56	78.56	16
60	depot	79.09				
61	depot	0	18.77	19.77	19.77	
61	13	20	20.55	21.55	21.55	16
61	depot	21.79	24.1	25.1	25.1	
61	18	25.62	25.62	26.62	26.62	16
61	depot	27.15	29.38	30.38	30.38	
61	14	30.57	30.57	31.57	31.57	16
61	depot	31.76	40.31	41.31	41.31	
61	15	41.4	41.4	42.4	42.4	16
61	depot	42.48	53.77	54.77	54.77	
61	18	55.3	55.3	56.3	56.3	16
61	depot	56.83	76.04	77.04	77.04	
61	18	77.56	77.56	78.56	78.56	16
61	depot	79.09				
62	depot	0	18.77	19.77	19.77	
62	13	20	21.55	22.55	22.55	16
62	depot	22.79	24.44	25.44	25.44	
62	16	25.95	25.95	26.95	26.95	16
62	depot	27.45	29.38	30.38	30.38	
62	14	30.57	30.57	31.57	31.57	16
62	depot	31.76	40.31	41.31	41.31	
62	17	41.81	41.81	42.81	42.81	16
62	depot	43.31	54.88	55.88	55.88	
62	17	56.38	56.38	57.38	57.38	16
62	depot	57.88				

Vehicle	Tour	Arr Time	Start Off/Onload	End Off/Onload	Depart	Delivery
63	depot	0	18.77	19.77	19.77	
63	16	20.27	20.27	21.27	21.27	12
63	depot	21.77	24.44	25.44	25.44	
63	16	25.95	25.95	26.95	26.95	12
63	depot	27.45	30.92	31.92	31.92	
63	14	32.1	32.1	33.1	33.1	12
63	depot	33.29	40.31	41.31	41.31	
63	15	41.4	41.4	42.4	42.4	12
63	depot	42.48				
64	depot	0	19.32	20.32	20.32	
64	13	20.55	22.55	23.55	23.55	12
64	depot	23.79	24.88	25.88	25.88	
64	16	26.39	26.39	27.39	27.39	12
64	depot	27.89	31.31	32.31	32.31	
64	17	32.81	32.81	33.81	33.81	12
64	depot	34.31	40.31	41.31	41.31	
64	17	41.81	41.81	42.81	42.81	12
64	depot	43.31	54.88	55.88	55.88	
64	17	56.38	56.38	57.38	57.38	7
64	depot	57.88				
65	depot	0	19.77	20.77	20.77	
65	17	21.27	21.27	22.27	22.27	12
65	depot	22.77	26.45	27.45	27.45	
65	16	27.95	27.95	28.95	28.95	12
65	depot	29.45	32.75	33.75	33.75	
65	17	34.25	34.25	35.25	35.25	12
65	depot	35.75	50.76	51.76	51.76	
65	18	52.29	52.29	53.29	53.29	12
65	depot	53.82	69.31	70.31	70.31	
65	15	70.4	70.4	71.4	71.4	10
65	depot	71.48				
66	depot	0	19.77	20.77	20.77	
66	16	21.43	21.43	22.43	22.43	8
66	depot	23.1	26.66	27.66	27.66	
66	16	28.32	28.32	29.32	29.32	8
66	depot	29.99	32.75	33.75	33.75	
66	14	34	34	35	35	7
66	depot	35.25	51.45	52.45	52.45	
66	18	53.15	53.15	54.15	54.15	8
66	depot	54.86	71.32	72.32	72.32	
66	18	73.03	73.03	74.03	74.03	8
66	depot	74.73				

Vehicle	Tour	Arr Time	Start Off/Onload	End Off/Onload	Depart	Delivery
67	depot	0	22.09	23.09	23.09	
67	13	23.41	23.55	24.55	24.55	5
67	14	25.01	25.01	26.01	26.01	3
67	depot	26.26	32.75	33.75	33.75	
67	15	33.86	33.86	34.86	34.86	8
67	depot	34.97	52.82	53.82	53.82	
67	18	54.53	54.53	55.53	55.53	8
67	depot	56.23	71.32	72.32	72.32	
67	18	73.03	73.03	74.03	74.03	8
67	depot	74.73	76.04	77.04	77.04	
67	18	77.74	77.74	78.74	78.74	7
67	depot	79.44				
68	depot	0	22.09	23.09	23.09	
68	14	23.34	23.34	24.34	24.34	8
68	depot	24.59	26.66	27.66	27.66	
68	14	27.9	27.9	28.9	28.9	8
68	depot	29.15	32.75	33.75	33.75	
68	17	34.42	34.42	35.42	35.42	8
68	depot	36.08	52.82	53.82	53.82	
68	15	53.93	53.93	54.93	54.93	8
68	depot	55.05				
69	depot	0	26.66	27.66	27.66	
69	16	28.16	28.16	29.16	29.16	4
69	depot	29.66	33.76	34.76	34.76	
69	17	35.26	35.26	36.26	36.26	4
69	depot	36.76				
70	depot	0	13.74	14.74	14.74	
70	23	15.24	24	25	25	16
70	depot	25.5	27.12	28.12	28.12	
70	22	28.62	28.62	29.62	29.62	16
70	depot	30.12	31.17	32.17	32.17	
70	20	32.29	32.29	33.29	33.29	16
70	depot	33.41	49.73	50.73	50.73	
70	22	51.23	51.23	52.23	52.23	16
70	depot	52.73	56.22	57.22	57.22	
70	20	57.34	57.34	58.34	58.34	10
70	depot	58.46	68.73	69.73	69.73	
70	21	69.9	69.9	70.9	70.9	10
70	depot	71.07				
71	depot	0	14.18	15.18	15.18	
71	20	15.3	24	25	25	16
71	depot	25.12	26.12	27.12	27.12	
71	20	27.24	27.24	28.24	28.24	16
71	depot	28.35	29.35	30.35	30.35	
71	22	30.85	30.85	31.85	31.85	16
71	depot	32.36	33.36	34.36	34.36	
71	21	34.52	34.52	35.52	35.52	16
71	depot	35.69	49.73	50.73	50.73	
71	22	51.23	52.23	53.23	53.23	16
71	depot	53.73	56.22	57.22	57.22	
71	22	57.73	57.73	58.73	58.73	7
71	depot	59.23				

Vehicle	Tour	Arr Time	Start Off/Onload	End Off/Onload	Depart	Delivery
72	depot	0	14.29	15.29	15.29	
72	19	15.46	24	25	25	16
72	depot	25.17	26.17	27.17	27.17	
72	23	27.67	27.67	28.67	28.67	16
72	depot	29.17	30.17	31.17	31.17	
72	23	31.67	31.67	32.67	32.67	16
72	depot	33.17	43.3	44.3	44.3	
72	24	44.81	44.81	45.81	45.81	16
72	depot	46.32	49.78	50.78	50.78	
72	19	50.95	50.95	51.95	51.95	16
72	depot	52.12				
73	depot	0	14.29	15.29	15.29	
73	24	15.79	24	25	25	12
73	depot	25.51	27.22	28.22	28.22	
73	22	28.72	29.62	30.62	30.62	12
73	depot	31.12	32.12	33.12	33.12	
73	23	33.62	33.62	34.62	34.62	11
73	depot	35.12	43.3	44.3	44.3	
73	24	44.81	44.81	45.81	45.81	12
73	depot	46.32				
74	depot	0	14.74	15.74	15.74	
74	24	16.25	24	25	25	12
74	depot	25.51	28.12	29.12	29.12	
74	19	29.29	29.34	30.34	30.34	12
74	depot	30.51	31.58	32.58	32.58	
74	24	33.08	33.08	34.08	34.08	12
74	depot	34.59	44.3	45.3	45.3	
74	24	45.81	45.81	46.81	46.81	12
74	depot	47.32	49.78	50.78	50.78	
74	22	51.28	53.23	54.23	54.23	12
74	depot	54.73				
75	depot	0	15.19	16.19	16.19	
75	23	16.69	24	25	25	12
75	depot	25.5	27.17	28.17	28.17	
75	19	28.34	28.34	29.34	29.34	12
75	depot	29.51	30.51	31.51	31.51	
75	23	32.01	32.01	33.01	33.01	12
75	depot	33.51	44.3	45.3	45.3	
75	21	45.47	45.47	46.47	46.47	12
75	depot	46.64	50.74	51.74	51.74	
75	21	51.9	51.9	52.9	52.9	12
75	depot	53.07				
76	depot	0	15.29	16.29	16.29	
76	21	16.51	24	25	25	8
76	depot	25.22	26.22	27.22	27.22	
76	24	27.9	27.9	28.9	28.9	8
76	depot	29.57	30.57	31.57	31.57	
76	23	32.24	32.24	33.24	33.24	8
76	depot	33.91	45.31	46.31	46.31	
76	24	46.99	46.99	47.99	47.99	8
76	depot	48.66	50.74	51.74	51.74	
76	19	51.96	51.96	52.96	52.96	8

Vehicle	Tour	Arr Time	Start Off/Onload	End Off/Onload	Depart	Delivery
77	depot	0	15.29	16.29	16.29	
77	21	16.51	24	25	25	8
77	depot	25.22	26.22	27.22	27.22	
77	24	27.9	27.9	28.9	28.9	8
77	depot	29.57	30.57	31.57	31.57	
77	23	32.24	32.67	33.67	33.67	8
77	depot	34.34	45.31	46.31	46.31	
77	24	46.99	46.99	47.99	47.99	7
77	depot	48.66	50.79	51.79	51.79	
77	19	52.01	52.96	53.96	53.96	1
77	depot	54.19				
78	depot	0	15.74	16.74	16.74	
78	23	17.41	24	25	25	8
78	depot	25.67	28.17	29.17	29.17	
78	24	29.85	29.85	30.85	30.85	8
78	depot	31.52	40.29	41.29	41.29	
78	22	41.96	41.96	42.96	42.96	8
78	depot	43.62	45.31	46.31	46.31	
78	20	46.47	46.47	47.47	47.47	8
78	depot	47.63	50.79	51.79	51.79	
78	22	52.45	54.23	55.23	55.23	8
78	depot	55.9				
79	depot	0	16.19	17.19	17.19	
79	22	17.69	24	25	25	4
79	depot	25.5	27.22	28.22	28.22	
79	23	28.72	28.72	29.72	29.72	4
79	depot	30.22	31.51	32.51	32.51	
79	23	33.01	33.01	34.01	34.01	4
79	depot	34.51	40.29	41.29	41.29	
79	20	41.41	41.41	42.41	42.41	4
79	depot	42.52	45.31	46.31	46.31	
79	21	46.48	46.48	47.48	47.48	4
79	depot	47.64				
80	depot	0	27.67	28.67	28.67	
80	30	29.2	48	49	49	16
80	depot	49.53	50.53	51.53	51.53	
80	28	52.03	52.03	53.03	53.03	16
80	depot	53.54	54.54	55.54	55.54	
80	29	56.04	56.04	57.04	57.04	16
80	depot	57.54	58.54	59.54	59.54	
80	28	70.82	70.82	71.82	71.82	16
80	depot	72.33	73.33	74.33	74.33	
80	27	84.69	84.69	85.69	85.69	16
80	depot	85.85	86.85	87.85	87.85	
80	26	87.97	87.97	88.97	88.97	2
80	depot	89.09				

Vehicle	Tour	Arr Time	Start Off/Onload	End Off/Onload	Depart	Delivery
81	depot	0	39.76	40.76	40.76	
81	25	40.96	48	49	49	16
81	depot	49.2	50.2	51.2	51.2	
81	26	51.32	51.32	52.32	52.32	16
81	depot	52.44	53.44	54.44	54.44	
81	30	54.97	54.97	55.97	55.97	16
81	depot	56.49	64.31	65.31	65.31	
81	30	65.84	65.84	66.84	66.84	16
81	depot	67.37	74.03	75.03	75.03	
81	26	75.14	75.14	76.14	76.14	16
81	depot	76.26	83.52	84.52	84.52	
81	27	84.69	84.69	85.69	85.69	14
81	depot	85.85				
82	depot	0	40.76	41.76	41.76	
82	28	42.26	48	49	49	16
82	depot	49.51	50.51	51.51	51.51	
82	28	52.01	52.01	53.01	53.01	16
82	depot	53.52	54.52	55.52	55.52	
82	29	56.02	56.02	57.02	57.02	16
82	depot	57.52	64.31	65.31	65.31	
82	29	65.81	65.81	66.81	66.81	16
82	depot	67.31	74.03	75.03	75.03	
82	26	75.14	75.14	76.14	76.14	16
82	depot	76.26				
83	depot	0	40.76	41.76	41.76	
83	30	42.28	48	49	49	12
83	depot	49.53	51.2	52.2	52.2	
83	25	52.41	52.71	53.71	53.71	12
83	depot	53.91	54.91	55.91	55.91	
83	28	56.42	56.42	57.42	57.42	12
83	depot	57.93	64.31	65.31	65.31	
83	30	65.84	65.84	66.84	66.84	12
83	depot	67.37	74.03	75.03	75.03	
83	25	75.23	75.23	76.23	76.23	0
83	28	76.56	76.56	77.56	77.56	3
83	depot	78.07				
84	depot	0	40.76	41.76	41.76	
84	28	42.26	48	49	49	12
84	depot	49.51	50.51	51.51	51.51	
84	25	51.71	51.71	52.71	52.71	12
84	depot	52.91	53.91	54.91	54.91	
84	29	55.41	55.41	56.41	56.41	12
84	depot	56.91	64.31	65.31	65.31	
84	29	65.81	65.81	66.81	66.81	12
84	depot	67.31	74.03	75.03	75.03	
84	27	75.19	75.19	76.19	76.19	12
84	depot	76.36				

Vehicle	Tour	Arr Time	Start Off/Onload	End Off/Onload	Depart	Delivery
85	depot	0	41.76	42.76	42.76	
85	25	42.96	49	50	50	12
85	depot	50.2	51.51	52.51	52.51	
85	25	52.71	53.71	54.71	54.71	9
85	depot	54.91	56.44	57.44	57.44	
85	29	57.94	57.94	58.94	58.94	12
85	depot	59.44	65.32	66.32	66.32	
85	30	66.84	66.84	67.84	67.84	12
85	depot	68.37				
86	depot	0	41.76	42.76	42.76	
86	28	43.43	49	50	50	8
86	depot	50.68	51.68	52.68	52.68	
86	28	53.35	53.35	54.35	54.35	8
86	depot	55.03	56.44	57.44	57.44	
86	30	58.15	58.15	59.15	59.15	8
86	depot	59.85	65.32	66.32	66.32	
86	30	67.02	67.02	68.02	68.02	7
86	depot	68.72	75.03	76.03	76.03	
86	27	76.25	76.25	77.25	77.25	8
86	depot	77.47				
87	depot	0	42.32	43.32	43.32	
87	30	44.03	48	49	49	8
87	depot	49.7	51.51	52.51	52.51	
87	30	53.21	53.21	54.21	54.21	8
87	depot	54.91	56.44	57.44	57.44	
87	29	58.11	58.19	59.19	59.19	8
87	depot	59.85	65.32	66.32	66.32	
87	29	66.98	66.98	67.98	67.98	7
87	depot	68.65	80.82	81.82	81.82	
87	27	82.04	82.04	83.04	83.04	8
87	depot	83.26				
88	depot	0	42.32	43.32	43.32	
88	28	44	49	50	50	8
88	depot	50.68	52.21	53.21	53.21	
88	26	53.36	53.36	54.36	54.36	8
88	depot	54.52	55.52	56.52	56.52	
88	29	57.19	57.19	58.19	58.19	8
88	depot	58.85	68.75	69.75	69.75	
88	26	69.91	69.91	70.91	70.91	8
88	depot	71.07	80.82	81.82	81.82	
88	27	82.04	82.04	83.04	83.04	8
88	depot	83.26				
89	depot	0	42.77	43.77	43.77	
89	25	43.97	50	51	51	4
89	depot	51.2	52.51	53.51	53.51	
89	29	54.01	54.01	55.01	55.01	4
89	depot	55.51	56.52	57.52	57.52	
89	29	58.02	58.02	59.02	59.02	4
89	depot	59.52	68.75	69.75	69.75	
89	26	69.87	69.87	70.87	70.87	4
89	depot	70.99	80.82	81.82	81.82	
89	27	81.98	81.98	82.98	82.98	4

Table C.2 Customer Delivery Schedules

Customer	CustLet	Veh	VehLet	Delivery	Arr	Offload	Dep
0	599	13	76	28	7.56	7.56	8.56
0	609	24	139	12	12.42	12.42	13.42
0	610	25	145	12	12.42	12.42	13.42
0	603	18	103	16	13.42	13.42	14.42
0	611	22	130	12	14.16	14.16	15.16
0	606	17	100	16	26.9	26.9	27.9
0	605	16	94	16	37.53	37.53	38.53
0	608	19	112	16	45.53	45.53	46.53
0	602	16	95	16	48.36	48.36	49.36
0	607	13	79	28	53.3	53.3	54.3
0	604	18	107	16	58.53	58.53	59.53
0	598	12	75	28	67.21	67.21	68.21
0	597	13	80	28	68.41	68.41	69.41
0	601	15	90	28	73.36	73.36	74.36
0	600	14	85	28	73.43	73.43	74.43
1	624	5	22	12	3.22	3.22	4.22
1	627	8	43	12	3.22	3.22	4.22
1	625	22	129	12	4.37	4.37	5.37
1	634	16	91	16	5.37	5.37	6.37
1	636	17	97	16	5.37	5.37	6.37
1	631	14	81	28	13.49	13.49	14.49
1	623	20	117	12	14.06	14.06	15.06
1	620	17	99	16	16.11	16.11	17.11
1	628	25	147	12	22.2	22.2	23.2
1	633	15	87	28	28.41	28.41	29.41
1	630	14	82	28	28.48	28.48	29.48
1	626	23	135	12	31.74	31.74	32.74
1	622	19	111	16	34.74	34.74	35.74
1	621	18	105	16	35.48	35.48	36.48
1	629	9	54	8	39.8	39.8	40.8
1	618	15	88	28	43.4	43.4	44.4
1	617	14	83	28	43.47	43.47	44.47
2	670	20	116	12	4.34	4.34	5.34
2	658	21	123	12	4.34	4.34	5.34
2	667	12	71	28	7.46	7.46	8.46
2	676	23	133	12	12.34	12.34	13.34
2	638	19	109	16	13.34	13.34	14.34
2	666	15	86	28	13.46	13.46	14.46
2	679	26	151	8	13.46	13.46	14.46
2	678	21	124	12	14.03	14.34	15.34
2	650	16	92	16	16.08	16.08	17.08
2	681	31	186	4	16.34	16.34	17.34
2	685	32	192	4	16.34	16.34	17.34
2	673	23	134	12	22.03	22.03	23.03
2	640	24	140	12	22.18	22.18	23.18
2	662	12	72	28	22.37	22.37	23.37
2	663	13	77	28	22.57	23.03	24.03
2	677	20	118	12	23.77	23.77	24.77
2	642	22	131	12	23.92	23.92	24.92
2	669	19	110	16	24.03	24.03	25.03
2	668	18	104	16	24.18	24.77	25.77

Customer	CustLet	Veh	VehLet	Delivery	Arr	Offload	Dep
2	664	26	153	8	25.37	25.37	26.37
2	654	31	188	4	26.03	26.03	27.03
2	655	32	195	4	26.03	26.03	27.03
2	653	16	93	16	26.77	26.77	27.77
2	647	24	142	12	31.86	31.86	32.86
2	648	25	148	12	31.92	31.92	32.92
2	659	12	73	28	37.28	37.28	38.28
2	649	26	154	8	37.28	37.28	38.28
2	637	17	101	16	37.66	37.66	38.66
2	660	13	78	28	37.94	38.28	39.28
2	665	11	68	12	40.8	40.8	41.8
2	646	23	136	12	41.45	41.45	42.45
2	644	24	143	12	41.55	41.55	42.55
2	645	25	149	12	41.6	41.6	42.6
2	675	18	106	16	46.2	46.2	47.2
2	643	23	137	12	51.14	51.14	52.14
2	674	12	74	28	52.2	52.2	53.2
2	657	3	12	85	54.86	54.86	56.86
2	639	19	113	16	56.29	56.29	57.29
2	672	15	89	28	58.35	58.35	59.35
2	671	14	84	28	58.42	58.42	59.42
2	651	16	96	16	59.12	59.12	60.12
2	682	4	20	17	67.41	67.41	69.41
2	656	2	6	82	70.99	70.99	78
3	727	6	29	12	3.52	3.52	4.52
3	728	7	36	12	3.52	3.52	4.52
3	722	3	8	85	5.07	5.07	7.07
3	723	4	15	85	5.07	5.07	7.07
3	764	9	50	12	5.52	5.52	6.52
3	731	10	57	12	5.52	7.07	8.07
3	732	11	64	12	5.52	7.07	8.07
3	689	5	23	12	9.95	9.95	10.95
3	692	8	44	12	9.95	9.95	10.95
3	690	6	30	12	10.55	10.55	11.55
3	691	7	37	12	10.55	10.95	11.95
3	724	0	0	85	11.08	11.08	13.08
3	761	9	51	12	12.55	12.55	13.55
3	762	10	58	12	14.1	14.1	15.1
3	695	11	65	12	14.1	14.1	15.1
3	721	3	9	85	15.2	15.2	17.2
3	688	4	16	85	15.2	15.2	17.2
3	720	5	24	12	16.98	16.98	17.98
3	757	8	45	12	16.98	17.2	18.2
3	755	6	31	12	17.59	17.59	18.59
3	756	7	38	12	17.98	17.98	18.98
3	758	9	52	12	19.59	19.59	20.59
3	759	10	59	12	21.14	21.14	22.14
3	726	11	66	12	21.14	21.14	22.14
3	735	5	25	12	24.02	24.02	25.02
3	704	8	46	12	24.23	24.23	25.23
3	736	6	32	12	24.63	24.63	25.63
3	687	2	2	85	25.07	25.07	27.07
3	718	3	10	85	31.07	31.07	33.07

Customer	CustLet	Veh	VehLet	Delivery	Arr	Offload	Dep
3	719	4	17	85	31.07	31.07	33.07
3	725	1	1	85	31.08	31.08	33.08
3	703	7	39	12	31.52	33.07	34.07
3	739	9	53	12	31.52	33.07	34.07
3	740	10	60	12	31.52	33.08	34.08
3	741	11	67	12	31.52	34.07	35.07
3	693	5	26	12	34.52	34.52	35.52
3	694	6	33	12	34.52	34.52	35.52
3	730	8	47	12	34.52	35.07	36.07
3	754	2	3	85	37.07	37.07	39.07
3	763	7	40	12	40.1	40.1	41.1
3	697	10	61	12	40.12	40.12	41.12
3	733	3	11	85	41.2	41.2	43.2
3	700	4	18	85	41.2	41.2	43.2
3	751	2	4	85	47.2	47.2	49.2
3	760	4	19	85	55.07	55.07	57.07
3	738	2	5	85	61.07	61.07	63.07
3	714	3	13	85	64.78	64.78	66.78
3	696	4	20	68	65.2	65.2	67.2
4	769	35	221	16	26.65	26.65	27.65
4	770	36	231	12	26.76	26.76	27.76
4	767	34	210	28	27.47	27.65	28.65
4	773	39	261	4	28.76	28.76	29.76
4	776	35	222	16	31.59	31.59	32.59
4	777	36	232	12	31.87	31.87	32.87
4	768	33	200	28	32.19	32.59	33.59
4	780	39	262	4	33.71	33.71	34.71
4	772	39	263	4	38.65	38.65	39.65
4	778	37	242	8	41.59	41.59	42.59
4	775	34	212	28	44.59	44.59	45.59
4	774	33	202	28	56.44	56.44	57.44
4	779	38	255	4	59.83	59.83	60.83
4	771	37	246	8	73.36	73.36	74.36
5	793	35	219	16	17.18	17.18	18.18
5	794	36	229	12	17.29	17.29	18.29
5	792	34	209	28	17.45	18.18	19.18
5	795	37	239	8	17.57	18.29	19.29
5	796	38	249	4	18.18	19.18	20.18
5	797	39	259	4	18.18	19.29	20.29
5	800	35	220	16	21.84	21.84	22.84
5	801	36	230	12	21.96	21.96	22.96
5	790	39	260	4	23.96	23.96	24.96
5	799	33	201	28	42.4	42.4	43.4
5	789	38	253	4	42.81	42.81	43.81
5	788	37	243	8	47	47	48
5	798	33	203	28	63.84	63.84	64.84
5	791	33	204	28	71.06	71.06	72.06
6	813	37	241	8	31.19	36	37
6	810	34	211	28	35.24	37	38
6	807	38	252	4	37.24	38	39
6	818	35	225	16	48.23	48.23	49.23
6	811	35	226	16	59.83	59.83	60.83
6	805	36	235	12	59.83	60.83	61.83

Customer	CustLet	Veh	VehLet	Delivery	Arr	Offload	Dep
6	808	39	265	4	60.83	61.83	62.83
6	814	38	256	4	64.77	64.77	65.77
6	812	36	236	12	65.77	65.77	66.77
6	815	39	266	4	66.77	66.77	67.77
6	806	37	245	8	66.95	67.77	68.77
6	803	34	215	28	75.09	75.09	76.09
6	816	33	205	28	78.47	78.47	79.47
6	809	33	206	28	86.06	86.06	87.06
7	819	33	199	28	13.36	24	25
7	830	37	240	8	24.3	25	26
7	831	38	250	4	24.43	26	27
7	832	38	251	4	31.85	31.85	32.85
7	828	35	223	16	36.99	36.99	37.99
7	829	36	233	12	37.27	37.99	38.99
7	821	35	224	16	42.84	42.84	43.84
7	822	36	234	12	45.23	45.23	46.23
7	825	39	264	4	45.88	46.23	47.23
7	824	38	254	4	48.06	48.06	49.06
7	827	34	213	28	58.1	58.1	59.1
7	823	37	244	8	60.76	60.76	61.76
7	820	34	214	28	66.89	66.89	67.89
7	826	34	216	28	83.28	83.28	84.28
8	841	48	326	16	17.22	17.22	18.22
8	839	47	319	16	23.58	23.58	24.58
8	838	42	285	28	31.57	31.57	32.57
8	837	59	406	4	41.46	41.46	42.46
8	846	47	322	11	49.18	49.18	50.18
9	855	54	367	8	12.02	12.02	13.02
9	853	55	376	8	29.56	29.56	30.56
9	849	50	343	12	51.57	51.57	52.57
9	847	43	295	28	70.28	70.28	71.28
9	848	44	302	19	70.28	70.28	71.28
10	869	40	269	28	16.31	16.31	17.31
10	875	46	311	16	16.77	16.77	17.77
10	876	47	318	16	16.77	16.77	17.77
10	870	41	276	28	16.8	17.31	18.31
10	889	54	368	8	17.76	17.77	18.77
10	859	52	354	12	20.09	20.09	21.09
10	891	42	284	28	21.21	21.21	22.21
10	885	58	397	4	22.1	22.1	23.1
10	864	43	291	28	22.44	22.44	23.44
10	860	53	361	12	22.88	22.88	23.88
10	862	55	375	8	23.45	23.45	24.45
10	892	57	389	8	24.45	24.45	25.45
10	868	46	312	16	24.66	24.66	25.66
10	861	40	270	28	26.45	26.45	27.45
10	890	41	277	28	27.38	27.38	28.38
10	857	50	340	12	28.92	28.92	29.92
10	858	51	347	12	29.31	29.31	30.31
10	879	52	355	12	30.75	30.75	31.75
10	880	53	362	12	30.75	30.75	31.75
10	865	58	398	4	30.75	30.75	31.75
10	886	59	405	4	31.76	31.76	32.76

Customer	CustLet	Veh	VehLet	Delivery	Arr	Offload	Dep
10	873	46	313	16	36.75	36.75	37.75
10	871	44	299	28	38.31	38.31	39.31
10	872	45	306	28	38.31	38.31	39.31
10	883	56	383	8	40.32	40.32	41.32
10	882	54	371	8	48.76	48.76	49.76
10	863	56	384	8	49.45	49.45	50.45
10	887	40	272	28	50.82	50.82	51.82
10	888	41	279	28	51.77	51.77	52.77
10	878	51	350	12	52.88	52.88	53.88
10	884	57	392	8	67.31	67.31	68.31
10	893	50	344	2	67.98	67.98	68.98
10	867	40	273	28	69.32	69.32	70.32
10	877	45	309	24	74.04	74.04	75.04
11	899	42	283	28	11.74	11.74	12.74
11	933	48	325	16	12.18	12.18	13.18
11	900	43	290	28	12.29	12.29	13.29
11	901	44	297	28	12.29	12.74	13.74
11	930	45	304	28	12.29	13.18	14.18
11	906	49	332	16	12.73	13.29	14.29
11	909	52	353	12	12.73	13.74	14.74
11	910	53	360	12	12.73	14.18	15.18
11	915	58	395	4	14.19	14.29	15.29
11	912	55	374	8	14.29	14.74	15.74
11	913	56	381	8	14.29	15.18	16.18
11	914	57	388	8	14.29	15.29	16.29
11	916	59	403	4	14.73	15.74	16.74
11	926	49	333	16	19.62	19.62	20.62
11	907	50	339	12	21.08	21.08	22.08
11	908	51	346	12	21.63	21.63	22.63
11	921	44	298	28	22.42	22.42	23.42
11	931	48	327	16	24.08	24.08	25.08
11	922	45	305	28	24.19	24.19	25.19
11	932	49	334	16	28.69	28.69	29.69
11	898	43	292	28	38.29	38.29	39.29
11	923	40	271	28	41.3	41.3	42.3
11	924	41	278	28	42.3	42.3	43.3
11	917	42	286	28	43.31	43.31	44.31
11	918	43	293	28	47.73	47.73	48.73
11	919	44	300	28	47.78	47.78	48.78
11	920	45	307	28	48.42	48.42	49.42
11	897	47	323	16	54.22	54.22	55.22
11	925	50	344	10	66.73	66.73	67.73
12	950	59	404	4	23.1	23.1	24.1
12	975	56	382	8	24.45	24.45	25.45
12	965	54	369	8	25.67	25.67	26.67
12	958	47	320	16	37.76	37.76	38.76
12	939	48	328	16	38.76	38.76	39.76
12	940	49	335	16	38.76	38.76	39.76
12	961	50	341	12	38.76	38.76	39.76
12	962	51	348	12	38.76	39.76	40.76
12	973	54	370	8	40.32	40.32	41.32
12	974	55	377	8	40.32	40.32	41.32
12	943	52	356	12	40.77	40.77	41.77

Customer	CustLet	Veh	VehLet	Delivery	Arr	Offload	Dep
12	968	57	390	8	41.33	41.33	42.33
12	944	53	363	12	41.77	41.77	42.77
12	969	58	399	4	41.77	41.77	42.77
12	937	46	314	16	43.12	43.12	44.12
12	938	47	321	16	44.13	44.13	45.13
12	959	48	329	16	45.13	45.13	46.13
12	960	49	336	16	45.13	45.13	46.13
12	941	50	342	12	45.13	45.13	46.13
12	942	51	349	12	46.13	46.13	47.13
12	970	59	407	4	47.13	47.13	48.13
12	963	52	357	12	48.14	48.14	49.14
12	966	55	378	8	48.81	48.81	49.81
12	964	53	364	12	49.15	49.15	50.15
12	948	57	391	8	49.77	49.77	50.77
12	957	46	315	16	49.9	49.9	50.9
12	951	48	330	16	52.88	52.88	53.88
12	953	42	287	28	54.44	54.44	55.44
12	954	43	294	28	62.31	62.31	63.31
12	955	44	301	28	62.31	62.31	63.31
12	956	45	308	28	62.31	62.31	63.31
12	949	46	316	16	66.75	66.75	67.75
12	967	56	385	8	67.31	67.31	68.31
12	952	41	280	28	72.03	72.03	73.03
12	945	42	288	28	72.03	72.03	73.03
12	971	40	274	28	78.82	78.82	79.82
12	972	41	281	22	81.52	81.52	82.52
13	977	60	411	16	19.55	19.55	20.55
13	978	61	417	16	20	20.55	21.55
13	979	62	423	16	20	21.55	22.55
13	981	64	437	12	20.55	22.55	23.55
13	984	67	455	5	23.41	23.55	24.55
14	985	68	461	8	23.34	23.34	24.34
14	986	67	455	3	25.01	25.01	26.01
14	987	68	462	8	27.9	27.9	28.9
14	990	61	419	16	30.57	30.57	31.57
14	991	62	425	16	30.57	30.57	31.57
14	989	63	432	12	32.1	32.1	33.1
14	992	66	451	7	34	34	35
15	998	67	456	8	33.86	33.86	34.86
15	994	61	420	16	41.4	41.4	42.4
15	999	63	433	12	41.4	41.4	42.4
15	995	60	415	16	53.91	53.91	54.91
15	996	68	465	8	53.93	53.93	54.93
15	993	65	447	10	70.4	70.4	71.4
16	1006	63	430	12	20.27	20.27	21.27
16	1007	66	449	8	21.43	21.43	22.43
16	1011	60	412	16	24.71	24.71	25.71
16	1002	62	424	16	25.95	25.95	26.95
16	1003	63	431	12	25.95	25.95	26.95
16	1004	64	438	12	26.39	26.39	27.39
16	1005	65	444	12	27.95	27.95	28.95
16	1010	69	467	4	28.16	28.16	29.16
16	1009	66	450	8	28.32	28.32	29.32

Customer	CustLet	Veh	VehLet	Delivery	Arr	Offload	Dep
16	1001	60	413	15	29.95	29.95	30.95
17	1018	65	443	12	21.27	21.27	22.27
17	1014	64	439	12	32.81	32.81	33.81
17	1015	65	445	12	34.25	34.25	35.25
17	1017	68	464	8	34.42	34.42	35.42
17	1024	69	468	4	35.26	35.26	36.26
17	1020	60	414	16	40.25	40.25	41.25
17	1022	62	426	16	41.81	41.81	42.81
17	1013	64	440	12	41.81	41.81	42.81
17	1021	62	427	16	56.38	56.38	57.38
17	1023	64	441	7	56.38	56.38	57.38
18	1025	61	418	16	25.62	25.62	26.62
18	1026	65	446	12	52.29	52.29	53.29
18	1027	66	452	8	53.15	53.15	54.15
18	1029	67	457	8	54.53	54.53	55.53
18	1032	61	421	16	55.3	55.3	56.3
18	1033	66	453	8	73.03	73.03	74.03
18	1028	67	458	8	73.03	73.03	74.03
18	1034	60	416	16	77.56	77.56	78.56
18	1035	61	422	16	77.56	77.56	78.56
18	1031	67	459	7	77.74	77.74	78.74
19	1039	72	485	16	15.46	24	25
19	1041	75	506	12	28.34	28.34	29.34
19	1040	74	500	12	29.29	29.34	30.34
19	1037	72	489	16	50.95	50.95	51.95
19	1043	76	515	8	51.96	51.96	52.96
19	1044	77	521	1	52.01	52.96	53.96
20	1050	71	479	16	15.3	24	25
20	1048	71	480	16	27.24	27.24	28.24
20	1049	70	475	16	32.29	32.29	33.29
20	1046	79	532	4	41.41	41.41	42.41
20	1045	78	526	8	46.47	46.47	47.47
20	1047	70	477	10	57.34	57.34	58.34
21	1055	76	511	8	16.51	24	25
21	1056	77	517	8	16.51	24	25
21	1053	71	482	16	34.52	34.52	35.52
21	1057	75	508	12	45.47	45.47	46.47
21	1058	79	533	4	46.48	46.48	47.48
21	1054	75	509	12	51.9	51.9	52.9
21	1059	70	478	10	69.9	69.9	70.9
22	1070	79	529	4	17.69	24	25
22	1071	70	474	16	28.62	28.62	29.62
22	1063	73	493	12	28.72	29.62	30.62
22	1066	71	481	16	30.85	30.85	31.85
22	1062	78	525	8	41.96	41.96	42.96
22	1064	70	476	16	51.23	51.23	52.23
22	1072	71	483	16	51.23	52.23	53.23
22	1065	74	503	12	51.28	53.23	54.23
22	1069	78	527	8	52.45	54.23	55.23
22	1061	71	484	7	57.73	57.73	58.73
23	1077	70	473	16	15.24	24	25
23	1078	75	505	12	16.69	24	25
23	1081	78	523	8	17.41	24	25

Customer	CustLet	Veh	VehLet	Delivery	Arr	Offload	Dep
23	1074	72	486	16	27.67	27.67	28.67
23	1076	79	530	4	28.72	28.72	29.72
23	1079	72	487	16	31.67	31.67	32.67
23	1082	75	507	12	32.01	32.01	33.01
23	1083	76	513	8	32.24	32.24	33.24
23	1073	77	519	8	32.24	32.67	33.67
23	1075	79	531	4	33.01	33.01	34.01
23	1080	73	494	11	33.62	33.62	34.62
24	1088	73	492	12	15.79	24	25
24	1089	74	499	12	16.25	24	25
24	1085	76	512	8	27.9	27.9	28.9
24	1086	77	518	8	27.9	27.9	28.9
24	1087	78	524	8	29.85	29.85	30.85
24	1093	74	501	12	33.08	33.08	34.08
24	1090	72	488	16	44.81	44.81	45.81
24	1091	73	495	12	44.81	44.81	45.81
24	1092	74	502	12	45.81	45.81	46.81
24	1094	76	514	8	46.99	46.99	47.99
24	1095	77	520	7	46.99	46.99	47.99
25	1098	81	541	16	40.96	48	49
25	1102	85	567	12	42.96	49	50
25	1104	89	591	4	43.97	50	51
25	1100	84	562	12	51.71	51.71	52.71
25	1099	83	555	12	52.41	52.71	53.71
25	1101	85	568	9	52.71	53.71	54.71
26	1109	81	542	16	51.32	51.32	52.32
26	1111	88	586	8	53.36	53.36	54.36
26	1106	89	594	4	69.87	69.87	70.87
26	1105	88	588	8	69.91	69.91	70.91
26	1108	81	545	16	75.14	75.14	76.14
26	1110	82	551	16	75.14	75.14	76.14
26	1107	80	540	2	87.97	87.97	88.97
27	1113	84	565	12	75.19	75.19	76.19
27	1115	86	577	8	76.25	76.25	77.25
27	1118	89	595	4	81.98	81.98	82.98
27	1116	87	583	8	82.04	82.04	83.04
27	1117	88	589	8	82.04	82.04	83.04
27	1119	80	539	16	84.69	84.69	85.69
27	1120	81	546	14	84.69	84.69	85.69
28	1123	82	547	16	42.26	48	49
28	1125	84	561	12	42.26	48	49
28	1127	86	573	8	43.43	49	50
28	1129	88	585	8	44	49	50
28	1122	82	548	16	52.01	52.01	53.01
28	1131	80	536	16	52.03	52.03	53.03
28	1121	86	574	8	53.35	53.35	54.35
28	1128	83	556	12	56.42	56.42	57.42
28	1124	80	538	16	70.82	70.82	71.82
28	1126	83	558	3	76.56	76.56	77.56
29	1136	89	592	4	54.01	54.01	55.01
29	1141	84	563	12	55.41	55.41	56.41
29	1139	82	549	16	56.02	56.02	57.02
29	1137	80	537	16	56.04	56.04	57.04

Customer	CustLet	Veh	VehLet	Delivery	Arr	Offload	Dep
29	1134	88	587	8	57.19	57.19	58.19
29	1142	85	569	12	57.94	57.94	58.94
29	1135	89	593	4	58.02	58.02	59.02
29	1133	87	581	8	58.11	58.19	59.19
29	1138	82	550	16	65.81	65.81	66.81
29	1140	84	564	12	65.81	65.81	66.81
29	1143	87	582	7	66.98	66.98	67.98
30	1145	80	535	16	29.2	48	49
30	1148	83	554	12	42.28	48	49
30	1152	87	579	8	44.03	48	49
30	1146	87	580	8	53.21	53.21	54.21
30	1150	81	543	16	54.97	54.97	55.97
30	1155	86	575	8	58.15	58.15	59.15
30	1149	81	544	16	65.84	65.84	66.84
30	1151	83	557	12	65.84	65.84	66.84
30	1153	85	570	12	66.84	66.84	67.84
30	1154	86	576	7	67.02	67.02	68.02

Appendix D: Benchmark Problems

Data Sets 1 – 9, 28, 31, 34 Vehicle Data

veh	# trips	capacity	speed	serv time	location coordinates		fixed cost	var cost
0	1	85	470	2	0	0	0	0.05
1	1	85	470	2	0	0	0	0.05
2	3	85	470	2	0	60	10	0.05
3	4	85	470	2	0	60	10	0.05
4	2	85	470	2	0	60	10	0.05
5	5	85	470	2	0	60	10	0.05
6	5	12	330	1	0	60	2	0.02
7	5	12	330	1	0	60	2	0.02
8	5	12	330	1	0	60	2	0.02
9	5	12	330	1	0	60	2	0.02
10	5	12	330	1	0	60	2	0.02
11	5	12	330	1	0	60	2	0.02
12	5	12	330	1	0	60	2	0.02
13	5	12	330	1	0	60	2	0.02
14	5	12	330	1	0	60	2	0.02
15	5	12	330	1	0	60	2	0.02
16	4	12	330	1	0	60	2	0.02

veh	DorH	AorG	AvTime	RL	DD	dep#	ldTime	unldTime
0	0	0	10	9000	1	0	4	2
1	0	0	20	9000	1	0	4	2
2	0	0	20	9000	0	1	4	2
3	0	0	0	9000	0	1	4	2
4	0	0	30	9000	0	1	4	2
5	0	0	0	9000	0	1	4	2
6	0	0	0	3000	0	1	2	1
7	0	0	0	3000	0	1	2	1
8	0	0	0	3000	0	1	2	1
9	0	0	0	3000	0	1	2	1
10	0	0	0	3000	0	1	2	1
11	0	0	0	3000	0	1	2	1
12	0	0	0	3000	0	1	2	1
13	0	0	0	3000	0	1	2	1
14	0	0	0	3000	0	1	2	1
15	0	0	0	3000	0	1	2	1
16	0	0	12	3000	0	1	2	1

Data Set 1

Air Force Multiple Trip Multiple Service with no hubs TDVRSP low delivery restrictions, low demand/cap

cust	locationCoor		demand	wMOGA	pMOGA	eTDD	TDD	priority
0	400	90	300	1	2	0	48	1
1	400	105	300	2	4	0	47	1
2	400	75	300	3	6	0	44	1
3	500	90	300	3	6	0	48	1
4	420	30	100	2	4	0	40	1
5	420	15	100	2	4	0	44	1
6	420	45	100	1	2	0	46	1
7	500	30	100	1	2	0	48	1

cust	CummulativeDemand	ETDD	TDD
0	69	0	21
0	142	0	30
0	219	0	40
0	300	0	48
1	70	0	18
1	150	0	32
1	215	0	39
1	300	0	47
2	85	0	18
2	150	0	24
2	205	0	37
2	300	0	44
3	86	0	21
3	142	0	30
3	215	0	40
3	300	0	48
4	40	0	26
4	100	0	40
5	55	0	30
5	100	0	44
6	50	0	23
6	100	0	46
7	40	0	32
7	100	0	48

Hub/Depot	WMOGA	WMOGG	PMOGA
APOD	6	Na	6

Data Set 2

Air Force Multiple Trip Multiple Service with no hubs TDVRSP low delivery restrictions, medium demand/cap

cust	locationCoor		demand	wMOGA	pMOGA	eTDD	TDD	priority
0	400	105	360	1	2	0	47	1
1	400	75	360	2	4	0	48	1
2	400	90	360	3	6	0	44	1
3	500	90	360	3	6	0	48	1
4	420	15	120	3	6	0	44	1
5	420	30	120	3	6	0	42	1
6	420	45	120	1	2	0	47	1
7	500	30	120	1	2	0	40	1

cust	CummulativeDemand	ETDD	TDD
0	75	0	20
0	165	0	28
0	270	0	38
0	360	0	47
1	80	0	18
1	170	0	34
1	254	0	40
1	360	0	48
2	102	0	18
2	176	0	26
2	173	0	38
2	360	0	44
3	100	0	22
3	184	0	28
3	262	0	40
3	360	0	48
4	60	0	15
4	120	0	44
5	58	0	33
5	120	0	42
6	84	0	18
6	120	0	47
7	96	0	30
7	120	0	40

Hub/Depot	WMOGA	WMOGG	PMOGA
APOD	6	Na	6

Data Set 3

Air Force Multiple Trip Multiple Service with no hubs TDVRSP low delivery restrictions, high demand/cap

cust	locationCoor		demand	wMOGA	pMOGA	eTDD	TDD	priority
0	400	90	400	1	2	0	47	0.8
1	400	105	400	2	4	0	47	1
2	400	75	400	3	6	0	46	0.9
3	500	90	400	3	6	0	48	1
4	420	30	150	1	2	0	44	0.7
5	420	15	150	3	6	0	42	1
6	420	45	150	1	2	0	47	0.6
7	500	30	150	3	6	0	40	1

cust	CummulativeDemand	ETDD	TDD
0	100	0	20
0	197	0	32
0	299	0	39
0	400	0	47
1	105	0	18
1	215	0	32
1	310	0	42
1	400	0	47
2	97	0	18
2	194	0	26
2	303	0	39
2	400	0	46
3	112	0	23
3	212	0	32
3	292	0	42
3	400	0	48
4	75	0	15
4	150	0	44
5	58	0	18
5	150	0	42
6	84	0	33
6	150	0	47
7	96	0	30
7	150	0	40

Hub/Depot	WMOGA	WMOGG	PMOGA
APOD	6	Na	6

Data Set 4

Air Force Multiple Trip Multiple Service with no hubs TDVRSP medium delivery restrictions, low demand/cap

cust	locationCoor		demand	wMOGA	pMOGA	eTDD	TDD	prior
0	400	90	315	1	3	0	48	1
1	400	105	315	2	4	0	46	1
2	400	75	315	2	4	0	40	1
3	500	90	315	3	6	0	38	1
4	420	15	105	3	6	0	45	1
5	420	30	105	2	4	0	44	1
6	420	45	105	2	4	0	42	1
7	500	30	105	1	1	0	48	1

cust	CummulativeDemand	ETDD	TDD
0	71	0	12
0	142	0	30
0	240	0	35
0	315	0	48
1	77	0	15
1	154	0	32
1	233	0	36
1	315	0	46
2	80	0	18
2	155	0	24
2	260	0	32
2	315	0	40
3	83	0	20
3	170	0	27
3	238	0	34
3	315	0	38
4	25	0	12
4	45	0	24
4	75	0	36
4	105	0	45
5	30	0	23
5	55	0	30
5	87	0	36
5	105	0	44
6	28	0	18
6	50	0	28
6	80	0	36
6	105	0	42
7	24	0	20
7	52	0	32
7	77	0	36
7	105	0	48

Hub/Depot	WMOGA	WMOGG	PMOGA
APOD	6	Na	6

Data Set 5

Air Force Multiple Trip Multiple Service with no hubs TDVRSP medium delivery restrictions, medium demand/cap

cust	locationCoor		demand	wMOGA	pMOGA	eTDD	TDD	prior
0	400	75	375	1	3	0	46	1
1	400	90	375	2	4	0	48	1
2	400	105	375	2	4	0	44	1
3	500	90	375	3	6	0	44	1
4	420	15	125	1	2	0	48	1
5	420	30	125	2	4	0	46	1
6	420	45	125	2	4	0	45	1
7	500	30	125	3	6	0	44	1

cust	CummulativeDemand	ETDD	TDD
0	90	0	12
0	185	0	29
0	280	0	36
0	375	0	46
1	100	0	16
1	180	0	30
1	280	0	35
1	375	0	48
2	95	0	16
2	185	0	24
2	275	0	32
2	375	0	44
3	80	0	18
3	185	0	28
3	285	0	34
3	375	0	44
4	25	0	18
4	55	0	30
4	90	0	36
4	125	0	48
5	24	0	17
5	56	0	26
5	84	0	34
5	125	0	46
6	22	0	15
6	50	0	24
6	85	0	36
6	125	0	45
7	24	0	13
7	52	0	26
7	84	0	34
7	125	0	44

Hub/Depot	WMOGA	WMOGG	PMOGA
APOD	6	Na	6

Data Set 6

Air Force Multiple Trip Multiple Service with no hubs TDVRSP medium delivery restrictions, high demand/cap

cust	locationCoor		demand	wMOGA	pMOGA	eTDD	TDD	prior
0	400	105	410	1	2	0	44	0.9
1	400	75	410	2	4	0	48	1
2	400	90	410	2	4	0	42	0.7
3	500	90	410	3	6	0	40	1
4	420	15	160	2	4	0	40	0.8
5	420	30	160	1	2	0	48	1
6	420	45	160	3	6	0	40	0.6
7	500	30	160	2	4	0	42	1

cust	CummulativeDemand	ETDD	TDD
0	88	0	14
0	206	0	32
0	292	0	35
0	410	0	44
1	93	0	12
1	197	0	32
1	297	0	36
1	410	0	48
2	102	0	18
2	182	0	24
2	298	0	33
2	410	0	42
3	111	0	18
3	189	0	26
3	279	0	34
3	410	0	40
4	35	0	18
4	87	0	26
4	125	0	33
4	160	0	40
5	36	0	22
5	80	0	28
5	130	0	36
5	160	0	48
6	40	0	18
6	82	0	26
6	118	0	34
6	160	0	40
7	42	0	20
7	78	0	23
7	122	0	35
7	160	0	42

Hub/Depot	WMOGA	WMOGG	PMOGA
APOD	6	Na	6

Data Set 7

Air Force Multiple Trip Multiple Service with no hubs TDVRSP high delivery restrictions, low demand/cap

cust	locationCoor		demand	wMOGA	pMOGA	eTDD	TDD	prior
0	400	90	330	1	2	0	48	1
1	400	105	330	1	2	0	47	1
2	400	75	330	2	3	0	44	1
3	500	90	330	3	6	0	48	1
4	420	15	110	1	2	0	44	1
5	420	30	110	1	2	0	48	1
6	420	45	110	2	4	0	42	1
7	500	30	110	3	6	0	48	1

cust	CummulativeDemand	ETDD	TDD
0	70	0	12
0	160	0	20
0	255	0	35
0	330	0	48
1	65	0	15
1	143	0	22
1	225	0	36
1	330	0	47
2	92	0	18
2	163	0	24
2	247	0	32
2	330	0	44
3	98	0	20
3	169	0	24
3	250	0	34
3	330	0	48
4	25	0	16
4	53	0	23
4	87	0	30
4	110	0	44
5	21	0	12
5	47	0	24
5	82	0	36
5	110	0	48
6	18	0	10
6	50	0	20
6	90	0	30
6	110	0	42
7	30	0	20
7	60	0	24
7	85	0	34
7	110	0	48

Hub/Depot	WMOGA	WMOGG	PMOGA
APOD	6	Na	6

Data Set 8

Air Force Multiple Trip Multiple Service with no hubs TDVRSP high delivery restrictions, medium demand/cap

cust	locationCoor		demand	wMOGA	pMOGA	eTDD	TDD	prior
0	400	75	380	1	2	0	46	1
1	400	90	380	1	2	0	48	1
2	400	105	380	2	4	0	42	1
3	500	90	380	3	6	0	48	1
4	420	15	120	1	2	0	48	1
5	420	30	120	2	4	0	46	1
6	420	45	120	3	6	0	44	1
7	500	30	120	1	1	0	48	1

cust	CummulativeDemand	ETDD	TDD
0	92	0	10
0	174	0	22
0	274	0	35
0	380	0	46
1	85	0	14
1	181	0	22
1	275	0	34
1	380	0	48
2	71	0	18
2	182	0	23
2	264	0	32
2	380	0	42
3	82	0	14
3	175	0	21
3	263	0	33
3	380	0	48
4	28	0	16
4	52	0	22
4	98	0	34
4	120	0	48
5	20	0	12
5	50	0	20
5	80	0	30
5	120	0	46
6	30	0	18
6	54	0	24
6	92	0	36
6	120	0	44
7	25	0	19
7	55	0	24
7	90	0	36
7	120	0	48

Hub/Depot	WMOGA	WMOGG	PMOGA
APOD	6	Na	6

Data Set 9

Air Force Multiple Trip Multiple Service with no hubs TDVRSP high delivery restrictions, high demand/cap

cust	locationCoor		demand	wMOGA	pMOGA	eTDD	TDD	prior
0	400	90	450	1	2	0	48	0.5
1	400	105	450	1	2	0	45	0.9
2	400	75	450	2	4	0	44	0.9
3	500	90	450	3	6	0	42	0.6
4	420	15	150	3	6	0	42	1
5	420	30	150	2	4	0	46	0.8
6	420	45	150	1	2	0	48	0.9
7	500	30	150	1	2	0	48	1

cust	CummulativeDemand	ETDD	TDD
0	114	0	14
0	226	0	23
0	352	0	34
0	450	0	48
1	92	0	12
1	195	0	22
1	330	0	32
1	450	0	45
2	128	0	18
2	221	0	24
2	341	0	32
2	450	0	44
3	140	0	20
3	228	0	24
3	340	0	34
3	450	0	42
4	42	0	8
4	95	0	18
4	125	0	30
4	150	0	42
5	26	0	12
5	74	0	24
5	108	0	32
5	150	0	46
6	30	0	18
6	66	0	24
6	110	0	36
6	150	0	48
7	32	0	16
7	72	0	22
7	108	0	34
7	150	0	48

Hub/Depot	WMOGA	WMOGG	PMOGA
APOD	6	Na	6

Data Set 10 – 18, 29, 32, 35 Vehicle Data

veh	# trips	capacity	speed	serv time	location coordinates		fixed cost	var cost
0	1	85	470	2	0	0	0	0.05
1	1	85	470	2	0	0	0	0.05
2	3	85	470	2	0	60	10	0.05
3	2	85	470	2	0	60	10	0.05
4	4	85	470	2	0	60	10	0.05
5	5	12	330	1	0	60	2	0.02
6	5	12	330	1	0	60	2	0.02
7	5	12	330	1	0	60	2	0.02
8	5	12	330	1	0	60	2	0.02
9	5	12	330	1	0	60	2	0.02
10	5	12	330	1	0	60	2	0.02
11	5	12	330	1	0	60	2	0.02
12	5	12	330	1	0	60	2	0.02
13	5	12	330	1	0	60	2	0.02
14	5	12	330	1	0	60	2	0.02
15	5	12	330	1	0	60	2	0.02
16	3	28	45	2	200	120	2	0.01
17	3	28	45	2	200	120	2	0.01
18	3	28	45	2	200	120	2	0.01
19	3	28	45	2	200	120	2	0.01
20	3	28	45	2	200	120	2	0.01
21	3	28	45	2	200	120	2	0.01
22	3	28	45	2	200	120	2	0.01
23	4	16	60	1	200	120	1	0.01
24	4	16	60	1	200	120	1	0.01
25	4	16	60	1	200	120	1	0.01
26	4	16	60	1	200	120	1	0.01
27	4	16	60	1	200	120	1	0.01
28	4	16	60	1	200	120	1	0.01
29	4	16	60	1	200	120	1	0.01
30	4	12	60	1	200	120	1	0.01
31	4	12	60	1	200	120	1	0.01
32	4	12	60	1	200	120	1	0.01
33	4	12	60	1	200	120	1	0.01
34	4	12	60	1	200	120	1	0.01
35	4	12	60	1	200	120	1	0.01
36	3	8	45	1	200	120	1	0.01
37	4	4	60	1	200	120	1	0.01
38	4	4	60	1	200	120	1	0.01
39	4	4	60	1	200	120	1	0.01
40	4	4	60	1	200	120	1	0.01

veh	DorH	AorG	AvTime	RL	DD	dep#	ldTime	unldTime
0	0	0	10	9000	1	0	4	2
1	0	0	20	9000	1	0	4	2
2	0	0	20	9000	0	1	4	2
3	0	0	30	9000	0	1	4	2
4	0	0	0	9000	0	1	4	2
5	0	0	0	3000	0	1	2	1
6	0	0	0	3000	0	1	2	1
7	0	0	0	3000	0	1	2	1
8	0	0	0	3000	0	1	2	1
9	0	0	0	3000	0	1	2	1
10	0	0	0	3000	0	1	2	1
11	0	0	12	3000	0	1	2	1
12	0	0	12	3000	0	1	2	1
13	0	0	12	3000	0	1	2	1
14	0	0	12	3000	0	1	2	1
15	0	0	12	3000	0	1	2	1
16	0	1	0	300	0	2	3	1
17	0	1	0	300	0	2	3	1
18	0	1	0	300	0	2	3	1
19	0	1	0	300	0	2	3	1
20	0	1	0	300	0	2	3	1
21	0	1	6	300	0	2	3	1
22	0	1	6	300	0	2	3	1
23	0	1	6	300	0	2	2	1
24	0	1	0	300	0	2	2	1
25	0	1	0	300	0	2	2	1
26	0	1	0	300	0	2	2	1
27	0	1	8	300	0	2	2	1
28	0	1	8	300	0	2	2	1
29	0	1	8	300	0	2	2	1
30	0	1	0	300	0	2	1	1
31	0	1	0	300	0	2	1	1
32	0	1	0	300	0	2	1	1
33	0	1	8	300	0	2	1	1
34	0	1	8	300	0	2	1	1
35	0	1	8	300	0	2	1	1
36	0	1	0	300	0	2	1	1
37	0	1	0	300	0	2	1	1
38	0	1	0	300	0	2	1	1
39	0	1	12	300	0	2	1	1
40	0	1	12	300	0	2	1	1

Data Set 10

Joint Multiple Trip Multiple Service with no hubs TDVRSP
low delivery restrictions, low demand/cap

cust	locationCoor		demand	wMOGA	wMOGG	pMOGA	eTDD	TDD	prior
0	400	75	400	1	3	2	0	48	1
1	400	90	400	2	3	4	0	47	1
2	400	105	400	3	3	6	0	44	1
3	500	90	400	3	3	6	0	48	1
4	420	15	200	0	2	0	0	40	1
5	420	30	200	0	2	0	0	44	1
6	420	45	200	0	1	0	0	46	1
7	500	30	200	0	1	0	0	48	1

cust	CummulativeDemand	ETDD	TDD
0	69	0	21
0	142	0	30
0	269	0	40
0	400	0	48
1	70	0	18
1	180	0	32
1	285	0	39
1	400	0	47
2	85	0	18
2	170	0	24
2	245	0	37
2	400	0	44
3	100	0	22
3	192	0	30
3	295	0	40
3	400	0	48
4	85	0	14
4	200	0	40
5	120	0	30
5	200	0	44
6	72	0	23
6	200	0	46
7	140	0	32
7	200	0	48

Hub/Depot	WMOGA	WMOGG	PMOGA
APOD	6	Na	6
SPOD	0	6	0

Data Set 11

Joint Multiple Trip Multiple Service with no hubs TDVRSP
low delivery restrictions, medium demand/cap

cust	locationCoor		demand	wMOGA	wMOGG	pMOGA	eTDD	TDD	prior
0	400	105	480	1	3	2	0	47	1
1	400	75	480	2	3	4	0	48	1
2	400	90	480	3	2	6	0	44	1
3	500	90	480	3	2	6	0	48	1
4	420	15	240	0	3	0	0	44	1
5	420	30	240	0	3	0	0	42	1
6	420	45	240	0	1	0	0	47	1
7	500	30	240	0	1	0	0	40	1

cust	CummulativeDemand	ETDD	TDD
0	175	0	20
0	245	0	28
0	360	0	38
0	480	0	47
1	150	0	18
1	220	0	34
1	314	0	40
1	480	0	48
2	102	0	18
2	176	0	26
2	273	0	38
2	480	0	44
3	136	0	22
3	284	0	28
3	362	0	40
3	480	0	48
4	95	0	15
4	240	0	44
5	168	0	33
5	240	0	42
6	84	0	18
6	240	0	47
7	186	0	30
7	240	0	40

Hub/Depot	WMOGA	WMOGG	PMOGA
APOD	**6**	**Na**	**6**
SPOD	**0**	**6**	**0**

Data Set 12

Joint Multiple Trip Multiple Service with no hubs TDVRSP low delivery restrictions, high demand/cap

cust	locationCoor		demand	wMOGA	wMOGG	pMOGA	eTDD	TDD	prior
0	400	90	550	1	3	2	0	47	1
1	400	105	500	2	2	4	0	47	0.7
2	400	75	550	3	3	6	0	46	1
3	500	90	600	3	2	6	0	48	0.9
4	420	30	300	0	1	0	0	44	1
5	420	15	250	0	3	0	0	42	1
6	420	45	300	0	1	0	0	47	0.8
7	500	30	250	0	3	0	0	40	0.5

cust	CummulativeDemand	ETDD	TDD
0	100	0	20
0	297	0	32
0	399	0	39
0	550	0	47
1	125	0	18
1	285	0	32
1	410	0	42
1	500	0	47
2	147	0	18
2	294	0	26
2	403	0	39
2	550	0	46
3	152	0	23
3	287	0	32
3	492	0	42
3	600	0	48
4	75	0	15
4	300	0	44
5	58	0	18
5	250	0	42
6	184	0	33
6	300	0	47
7	156	0	30
7	250	0	40

Hub/Depot	WMOGA	WMOGG	PMOGA
APOD	6	Na	6
SPOD	0	6	0

Data Set 13

Joint Multiple Trip Multiple Service with no hubs TDVRSP medium delivery restrictions, low demand/cap

cust	locationCoor		demand	wMOGA	wMOGG	pMOGA	eTDD	TDD	prior
0	400	90	420	1	3	3	0	48	1
1	400	105	420	2	2	4	0	46	1
2	400	75	440	2	2	4	0	40	1
3	500	90	400	3	1	6	0	38	1
4	420	15	210	0	3	0	0	45	1
5	420	30	210	0	2	0	0	44	1
6	420	45	220	0	2	0	0	42	1
7	500	30	200	0	1	0	0	48	1

cust	CummulativeDemand	ETDD	TDD
0	71	0	12
0	242	0	30
0	340	0	35
0	420	0	48
1	77	0	15
1	254	0	32
1	333	0	36
1	420	0	46
2	120	0	18
2	215	0	24
2	290	0	32
2	440	0	40
3	93	0	20
3	190	0	27
3	298	0	34
3	400	0	38
4	50	0	12
4	110	0	24
4	165	0	36
4	210	0	45
5	80	0	23
5	155	0	30
5	187	0	36
5	210	0	44
6	68	0	18
6	140	0	28
6	180	0	36
6	220	0	42
7	64	0	20
7	142	0	32
7	177	0	36
7	200	0	48

Hub/Depot	WMOGA	WMOGG	PMOGA
APOD	6	Na	6
SPOD	0	6	0

Data Set 14

Joint Multiple Trip Multiple Service with no hubs TDVRSP medium delivery restrictions, medium demand/cap

cust	locationCoor		demand	wMOGA	wMOGG	pMOGA	eTDD	TDD	prior
0	400	75	500	1	2	3	0	46	1
1	400	90	600	2	3	4	0	48	1
2	400	105	400	2	2	4	0	44	1
3	500	90	500	3	1	6	0	44	1
4	420	15	250	0	1	0	0	48	1
5	420	30	300	0	2	0	0	46	1
6	420	45	250	0	2	0	0	45	1
7	500	30	200	0	3	0	0	44	1

cust	CummulativeDemand	ETDD	TDD
0	110	0	12
0	245	0	29
0	380	0	36
0	500	0	46
1	130	0	16
1	280	0	30
1	480	0	35
1	600	0	48
2	95	0	16
2	185	0	24
2	275	0	32
2	400	0	44
3	80	0	18
3	195	0	28
3	385	0	34
3	500	0	44
4	75	0	18
4	155	0	30
4	190	0	36
4	250	0	48
5	84	0	17
5	156	0	26
5	194	0	34
5	300	0	46
6	90	0	15
6	130	0	24
6	185	0	36
6	250	0	45
7	54	0	13
7	112	0	26
7	174	0	34
7	200	0	44

Hub/Depot	WMOGA	WMOGG	PMOGA
APOD	6	Na	6
SPOD	0	6	0

Data Set 15

Joint Multiple Trip Multiple Service with no hubs TDVRSP medium delivery restrictions, high demand/cap

cust	locationCoor		demand	wMOGA	wMOGG	pMOGA	eTDD	TDD	prior
0	400	105	550	1	2	2	0	44	1
1	400	75	560	2	2	4	0	48	0.9
2	400	90	560	2	3	4	0	42	1
3	500	90	550	3	1	6	0	40	0.8
4	420	15	310	0	2	0	0	40	0.8
5	420	30	300	0	1	0	0	48	0.8
6	420	45	300	0	3	0	0	40	0.8
7	500	30	290	0	2	0	0	42	1

cust	CummulativeDemand	ETDD	TDD
0	138	0	14
0	286	0	32
0	392	0	35
0	550	0	44
1	123	0	12
1	267	0	32
1	397	0	36
1	560	0	48
2	102	0	18
2	262	0	24
2	398	0	33
2	560	0	42
3	141	0	18
3	289	0	26
3	369	0	34
3	550	0	40
4	35	0	18
4	187	0	26
4	225	0	33
4	310	0	40
5	66	0	22
5	130	0	28
5	230	0	36
5	300	0	48
6	40	0	18
6	152	0	26
6	188	0	34
6	300	0	40
7	42	0	20
7	118	0	23
7	202	0	35
7	290	0	42

Hub/Depot	WMOGA	WMOGG	PMOGA
APOD	6	Na	6
SPOD	0	6	0

Data Set 16

Joint Multiple Trip Multiple Service with no hubs TDVRSP
high delivery restrictions, low demand/cap

cust	locationCoor		demand	wMOGA	wMOGG	pMOGA	eTDD	TDD	prior
0	400	90	420	1	3	2	0	48	1
1	400	105	420	1	2	2	0	47	1
2	400	75	460	2	1	3	0	44	1
3	500	90	460	3	1	6	0	48	1
4	420	15	220	0	1	0	0	44	1
5	420	30	230	0	1	0	0	48	1
6	420	45	210	0	2	0	0	42	1
7	500	30	220	0	3	0	0	48	1

cust	CummulativeDemand	ETDD	TDD
0	90	0	12
0	190	0	20
0	355	0	35
0	420	0	48
1	75	0	15
1	183	0	22
1	295	0	36
1	420	0	47
2	112	0	18
2	213	0	24
2	347	0	32
2	460	0	44
3	128	0	20
3	249	0	24
3	350	0	34
3	460	0	48
4	45	0	16
4	93	0	23
4	167	0	30
4	220	0	44
5	41	0	12
5	97	0	24
5	152	0	36
5	230	0	48
6	18	0	10
6	95	0	20
6	180	0	30
6	210	0	42
7	40	0	20
7	80	0	24
7	125	0	34
7	210	0	48

Hub/Depot	WMOGA	WMOGG	PMOGA
APOD	6	Na	6
SPOD	0	6	0

Data Set 17

Joint Multiple Trip Multiple Service with no hubs TDVRSP high delivery restrictions, medium demand/cap

cust	locationCoor		demand	wMOGA	wMOGG	pMOGA	eTDD	TDD	prior
0	400	75	490	1	3	2	0	46	1
1	400	90	490	1	1	2	0	48	1
2	400	105	500	2	2	4	0	42	1
3	500	90	520	3	1	6	0	48	1
4	420	15	250	0	1	0	0	48	1
5	420	30	240	0	2	0	0	46	1
6	420	45	260	0	3	0	0	44	1
7	500	30	250	0	1	0	0	48	1

cust	CummulativeDemand	ETDD	TDD
0	102	0	10
0	244	0	22
0	374	0	35
0	490	0	46
1	125	0	14
1	251	0	22
1	375	0	34
1	510	0	48
2	99	0	18
2	262	0	23
2	364	0	32
2	500	0	42
3	82	0	14
3	275	0	21
3	373	0	33
3	520	0	48
4	88	0	16
4	122	0	22
4	198	0	34
4	250	0	48
5	50	0	12
5	110	0	20
5	180	0	30
5	270	0	46
6	70	0	18
6	144	0	24
6	192	0	36
6	270	0	44
7	95	0	14
7	155	0	24
7	190	0	36
7	270	0	48

Hub/Depot	WMOGA	WMOGG	PMOGA
APOD	6	Na	6
SPOD	0	6	0

Data Set 18

Joint Multiple Trip Multiple Service with no hubs TDVRSP
high delivery restrictions, high demand/cap

cust	locationCoor		demand	wMOGA	wMOGG	pMOGA	eTDD	TDD	prior
0	400	90	600	1	1	2	0	48	1
1	400	105	600	1	2	2	0	45	0.9
2	400	75	600	2	3	4	0	44	0.8
3	500	90	600	3	1	6	0	42	1
4	420	15	300	0	3	0	0	42	0.7
5	420	30	300	0	2	0	0	46	0.5
6	420	45	300	0	1	0	0	48	1
7	500	30	300	0	1	0	0	48	1

cust	CummulativeDemand	ETDD	TDD
0	114	0	14
0	286	0	23
0	452	0	34
0	600	0	48
1	92	0	12
1	265	0	22
1	430	0	32
1	600	0	45
2	128	0	18
2	291	0	24
2	441	0	32
2	600	0	44
3	140	0	20
3	288	0	24
3	440	0	34
3	600	0	42
4	42	0	8
4	145	0	18
4	185	0	30
4	300	0	42
5	36	0	12
5	174	0	24
5	198	0	32
5	300	0	46
6	90	0	18
6	126	0	24
6	190	0	36
6	300	0	48
7	52	0	16
7	172	0	22
7	228	0	34
7	300	0	48

Hub/Depot	WMOGA	WMOGG	PMOGA
APOD	6	Na	6
SPOD	0	6	0

Data Set 19 – 27, 30, 33, 36, 37, 38, 39 Vehicle Data

veh	# trips	capacity	speed	serv time	location coordinates		fixed cost	var cost
0	1	85	470	2	0	0	0	0.05
1	1	85	470	2	0	0	0	0.05
2	6	85	470	2	0	60	10	0.05
3	7	85	470	2	0	60	10	0.05
4	7	85	470	2	0	60	10	0.05
5	7	12	330	1	0	60	2	0.02
6	7	12	330	1	0	60	2	0.02
7	7	12	330	1	0	60	2	0.02
8	7	12	330	1	0	60	2	0.02
9	7	12	330	1	0	60	2	0.02
10	7	12	330	1	0	60	2	0.02
11	7	12	330	1	0	60	2	0.02
12	5	28	45	2	200	120	1	0.01
13	5	28	45	2	200	120	1	0.01
14	5	28	45	2	200	120	1	0.01
15	5	28	45	2	200	120	1	0.01
16	6	16	60	1	200	120	1	0.01
17	6	16	60	1	200	120	1	0.01
18	6	16	60	1	200	120	1	0.01
19	6	16	60	1	200	120	1	0.01
20	6	12	60	1	200	120	1	0.01
21	6	12	60	1	200	120	1	0.01
22	6	12	60	1	200	120	1	0.01
23	6	12	60	1	200	120	1	0.01
24	6	12	60	1	200	120	1	0.01
25	6	12	60	1	200	120	1	0.01
26	6	8	45	1	200	120	0.5	0.01
27	7	4	60	1	200	120	0.5	0.01
28	7	4	60	1	200	120	0.5	0.01
29	7	4	60	1	200	120	0.5	0.01
30	7	4	60	1	200	120	0.5	0.01
31	7	4	60	1	200	120	0.5	0.01
32	7	4	60	1	200	120	0.5	0.01
33	10	28	45	2	400	90	1	0.01
34	10	28	45	2	400	90	1	0.01
35	10	16	60	1	400	90	1	0.01
36	10	12	60	1	400	90	1	0.01
37	10	8	45	1	400	90	0.5	0.01
38	10	4	60	1	400	90	0.5	0.01
39	10	4	60	1	400	90	0.5	0.01
40	7	28	45	2	500	90	1	0.01
41	7	28	45	2	500	90	1	0.01
42	7	28	45	2	500	90	1	0.01
43	7	28	45	2	500	90	1	0.01

veh	# trips	capacity	speed	serv time	location coordinates		fixed cost	var cost
44	7	28	45	2	500	90	1	0.01
45	7	28	45	2	500	90	1	0.01
46	7	16	60	1	500	90	1	0.01
47	7	16	60	1	500	90	1	0.01
48	7	16	60	1	500	90	1	0.01
49	7	16	60	1	500	90	1	0.01
50	7	12	60	1	500	90	1	0.01
51	7	12	60	1	500	90	1	0.01
52	7	12	60	1	500	90	1	0.01
53	7	12	60	1	500	90	1	0.01
54	7	8	45	1	500	90	0.5	0.01
55	7	8	45	1	500	90	0.5	0.01
56	7	8	45	1	500	90	0.5	0.01
57	7	8	45	1	500	90	0.5	0.01
58	8	4	60	1	500	90	0.5	0.01
59	8	4	60	1	500	90	0.5	0.01
60	6	16	60	1	600	105	1	0.01
61	6	16	60	1	600	105	1	0.01
62	7	16	60	1	600	105	1	0.01
63	7	12	60	1	600	105	1	0.01
64	6	12	60	1	600	105	1	0.01
65	6	12	60	1	600	105	1	0.01
66	6	8	45	1	600	105	0.5	0.01
67	6	8	45	1	600	105	0.5	0.01
68	6	8	45	1	600	105	0.5	0.01
69	6	4	60	1	600	105	0.5	0.01
70	6	16	60	1	600	90	1	0.01
71	6	16	60	1	600	90	1	0.01
72	7	16	60	1	600	90	1	0.01
73	7	12	60	1	600	90	1	0.01
74	6	12	60	1	600	90	1	0.01
75	6	12	60	1	600	90	1	0.01
76	6	8	45	1	600	90	0.5	0.01
77	6	8	45	1	600	90	0.5	0.01
78	6	8	45	1	600	90	0.5	0.01
79	6	4	60	1	600	90	0.5	0.01
80	6	16	60	1	600	75	1	0.01
81	6	16	60	1	600	75	1	0.01
82	7	16	60	1	600	75	1	0.01
83	7	12	60	1	600	75	1	0.01
84	6	12	60	1	600	75	1	0.01
85	6	12	60	1	600	75	1	0.01
86	6	8	45	1	600	75	0.5	0.01
87	6	8	45	1	600	75	0.5	0.01
88	6	8	45	1	600	75	0.5	0.01
89	6	4	60	1	600	75	0.5	0.01

veh	DorH	AorG	AvTime	RL	DD	dep#	ldTime	unldTime
0	0	0	10	9000	1	0	4	2
1	0	0	30	9000	1	0	4	2
2	0	0	20	9000	0	1	4	2
3	0	0	0	9000	0	1	4	2
4	0	0	0	9000	0	1	4	2
5	0	0	0	3000	0	1	2	1
6	0	0	0	3000	0	1	2	1
7	0	0	0	3000	0	1	2	1
8	0	0	0	3000	0	1	2	1
9	0	0	0	3000	0	1	2	1
10	0	0	0	3000	0	1	2	1
11	0	0	0	3000	0	1	2	1
12	0	1	0	300	0	2	3	1
13	0	1	0	300	0	2	3	1
14	0	1	6	300	0	2	3	1
15	0	1	6	300	0	2	3	1
16	0	1	0	300	0	2	2	1
17	0	1	0	300	0	2	2	1
18	0	1	8	300	0	2	2	1
19	0	1	8	300	0	2	2	1
20	0	1	0	300	0	2	1	1
21	0	1	0	300	0	2	1	1
22	0	1	0	300	0	2	1	1
23	0	1	8	300	0	2	1	1
24	0	1	8	300	0	2	1	1
25	0	1	8	300	0	2	1	1
26	0	1	8	300	0	2	1	1
27	0	1	0	300	0	2	1	1
28	0	1	0	300	0	2	1	1
29	0	1	0	300	0	2	1	1
30	0	1	12	300	0	2	1	1
31	0	1	12	300	0	2	1	1
32	0	1	12	300	0	2	1	1
33	1	1	0	300	0	0	2	1
34	1	1	0	300	0	0	2	1
35	1	1	0	300	0	0	1	1
36	1	1	0	300	0	0	1	1
37	1	1	0	300	0	0	1	1
38	1	1	0	300	0	0	1	1
39	1	1	0	300	0	0	1	1
40	2	1	8	300	0	0	2	1
41	2	1	8	300	0	0	2	1
42	2	1	0	300	0	0	2	1
43	2	1	0	300	0	0	2	1
44	2	1	0	300	0	0	2	1
45	2	1	0	300	0	0	2	1

veh	DorH	AorG	AvTime	RL	DD	dep#	ldTime	unldTime
46	2	1	12	300	0	0	1	1
47	2	1	12	300	0	0	1	1
48	2	1	0	300	0	0	1	1
49	2	1	0	300	0	0	1	1
50	2	1	18	300	0	0	1	1
51	2	1	18	300	0	0	1	1
52	2	1	0	300	0	0	1	1
53	2	1	0	300	0	0	1	1
54	2	1	0	300	0	0	1	1
55	2	1	0	300	0	0	1	1
56	2	1	0	300	0	0	1	1
57	2	1	0	300	0	0	1	1
58	2	1	0	300	0	0	1	1
59	2	1	0	300	0	0	1	1
60	3	1	12	300	0	0	1	1
61	3	1	0	300	0	0	1	1
62	3	1	0	300	0	0	1	1
63	3	1	12	300	0	0	1	1
64	3	1	0	300	0	0	1	1
65	3	1	0	300	0	0	1	1
66	3	1	12	300	0	0	1	1
67	3	1	0	300	0	0	1	1
68	3	1	0	300	0	0	1	1
69	3	1	0	300	0	0	1	1
70	4	1	10	300	0	0	1	1
71	4	1	0	300	0	0	1	1
72	4	1	0	300	0	0	1	1
73	4	1	10	300	0	0	1	1
74	4	1	0	300	0	0	1	1
75	4	1	0	300	0	0	1	1
76	4	1	10	300	0	0	1	1
77	4	1	0	300	0	0	1	1
78	4	1	0	300	0	0	1	1
79	4	1	0	300	0	0	1	1
80	5	1	15	300	0	0	1	1
81	5	1	0	300	0	0	1	1
82	5	1	0	300	0	0	1	1
83	5	1	15	300	0	0	1	1
84	5	1	0	300	0	0	1	1
85	5	1	0	300	0	0	1	1
86	5	1	15	300	0	0	1	1
87	5	1	0	300	0	0	1	1
88	5	1	0	300	0	0	1	1
89	5	1	0	300	0	0	1	1

Data Set 19

Joint Multiple Trip Multiple Service with hubs TDVRSP
low delivery restrictions, low demand/cap

cust	location	Coor	demand	WMOGA,	WMOGG,	PMOGA	eTDD	TDD	hub	tier	prior
0	400	75	300	1	3	2	0	96	0	0	1
1	400	90	300	2	3	4	0	96	0	0	1
2	400	105	800	3	3	6	0	96	1	0	1
3	500	90	1800	3	3	6	0	96	2	0	1
4	450	60	200	0	2	0	0	96	0	1	1
5	450	90	200	0	2	0	0	96	0	1	1
6	450	120	200	0	1	0	0	96	0	1	1
7	480	60	200	0	1	0	0	96	0	1	1
8	520	100	75	0	1	0	0	48	0	2	1
9	520	80	75	0	2	0	0	72	0	2	1
10	600	105	550	0	3	0	0	96	3	2	1
11	600	90	550	0	3	0	0	96	4	2	1
12	600	75	550	0	3	0	0	96	5	2	1
13	610	115	65	0	1	0	0	24	0	3	1
14	605	95	70	0	2	0	0	36	0	3	1
15	600	110	70	0	2	0	0	72	0	3	1
16	630	107	115	0	3	0	0	30	0	3	1
17	630	105	115	0	3	0	0	66	0	3	1
18	630	95	115	0	3	0	0	96	0	3	1
19	610	92	65	0	1	0	0	70	0	4	1
20	605	85	70	0	2	0	0	96	0	4	1
21	600	80	70	0	3	0	0	96	0	4	1
22	630	92	115	0	1	0	0	96	0	4	1
23	630	90	115	0	3	0	0	36	0	4	1
24	630	85	115	0	3	0	0	90	0	4	1
25	610	82	65	0	1	0	0	96	0	5	1
26	605	70	70	0	2	0	0	96	0	5	1
27	600	65	70	0	3	0	0	96	0	5	1
28	630	80	115	0	2	0	0	72	0	5	1
29	630	75	115	0	3	0	0	96	0	5	1
30	630	65	115	0	3	0	0	68	0	5	1

Hub/D	wMOGA	wMOGG	PMOGA
APOD	6	6	6
SPOD	0	6	0
1	0	3	0
2	0	4	0
3	0	3	0
4	0	3	0
5	0	3	0

cust	CummulativeDemand	ETDD	TDD
0	69	0	36
0	142	0	60
0	219	0	84
0	300	0	96
1	70	0	36
1	150	0	60
1	215	0	84
1	300	0	96
2	800	0	96
3	1800	0	96
4	90	0	36
4	200	0	96
5	110	0	60
5	200	0	96
6	100	0	36
6	200	0	96
7	80	0	60
7	200	0	96
8	75	0	48
9	75	0	72
10	550	0	96
11	550	0	96
12	550	0	96
13	65	0	24
14	70	0	36
15	70	0	72
16	115	0	30
17	115	0	66
18	115	0	96
19	65	0	70
20	70	0	96
21	70	0	96
22	115	0	96
23	115	0	36
24	115	0	90
25	65	0	96
26	70	0	96
27	70	0	96
28	115	0	72
29	115	0	96
30	115	0	68

Data Set 20

Joint Multiple Trip Multiple Service with hubs TDVRSP
low delivery restrictions, medium demand/cap

cust	locationCoor		demand	WMOGA,WMOGG,PMOGA			eTDD	TDD	hub	tier	prior
0	400	75	360	1	3	2	0	96	0	0	1
1	400	90	370	2	3	4	0	96	0	0	1
2	400	105	1000	3	3	4	0	96	1	0	1
3	500	90	2300	3	3	6	0	96	2	0	1
4	450	15	240	0	2	0	0	92	0	1	1
5	450	60	260	0	2	0	0	96	0	1	1
6	450	90	240	0	1	0	0	92	0	1	1
7	480	120	260	0	1	0	0	96	0	1	1
8	520	60	220	0	1	0	0	44	0	2	1
9	520	80	210	0	2	0	0	68	0	2	1
10	600	105	580	0	3	0	0	96	3	2	1
11	600	90	580	0	3	0	0	96	4	2	1
12	600	75	640	0	3	0	0	96	5	2	1
13	610	115	75	0	1	0	0	26	0	3	1
14	605	95	125	0	2	0	0	30	0	3	1
15	600	110	125	0	2	0	0	72	0	3	1
16	630	107	85	0	3	0	0	36	0	3	1
17	630	105	85	0	3	0	0	72	0	3	1
18	630	95	85	0	3	0	0	90	0	3	1
19	610	92	75	0	1	0	0	66	0	4	1
20	605	85	115	0	2	0	0	80	0	4	1
21	600	80	115	0	3	0	0	40	0	4	1
22	630	92	125	0	1	0	0	72	0	4	1
23	630	90	75	0	3	0	0	42	0	4	1
24	630	85	75	0	3	0	0	96	0	4	1
25	610	82	75	0	1	0	0	84	0	5	1
26	605	70	90	0	2	0	0	88	0	5	1
27	600	65	70	0	3	0	0	92	0	5	1
28	630	80	125	0	2	0	0	64	0	5	1
29	630	75	135	0	3	0	0	90	0	5	1
30	630	65	145	0	3	0	0	60	0	5	1

Hub/D	wMOGA	wMOGG	PMOGA
APOD	6	6	6
SPOD	0	6	0
1	0	3	0
2	0	4	0
3	0	3	0
4	0	3	0
5	0	3	0

cust	CummulativeDemand	ETDD	TDD
0	75	0	30
0	135	0	66
0	230	0	80
0	360	0	96
1	55	0	24
1	145	0	54
1	230	0	76
1	370	0	96
2	1000	0	96
3	2300	0	96
4	120	0	44
4	240	0	92
5	130	0	54
5	260	0	96
6	140	0	40
6	240	0	92
7	90	0	54
7	260	0	96
8	220	0	44
9	210	0	68
10	580	0	96
11	580	0	96
12	680	0	96
13	75	0	26
14	125	0	30
15	125	0	72
16	85	0	36
17	85	0	72
18	85	0	90
19	75	0	66
20	115	0	80
21	115	0	40
22	125	0	72
23	75	0	42
24	75	0	96
25	75	0	84
26	100	0	88
27	70	0	92
28	135	0	64
29	145	0	90
30	155	0	60

Data Set 21

Joint Multiple Trip Multiple Service with hubs TDVRSP
low delivery restrictions, high demand/cap

cust	locationCoor		demand	WMOGA,WMOGG,PMOGA			eTDD	TDD	hub	tier	prior
0	400	75	420	1	3	2	0	96	0	0	0.9
1	400	90	390	2	3	4	0	96	0	0	0.8
2	400	105	1180	3	3	6	0	96	1	0	1
3	500	90	2500	3	3	6	0	96	2	0	1
4	450	60	260	0	2	0	0	92	0	1	0.8
5	450	90	340	0	2	0	0	96	0	1	0.8
6	450	120	300	0	1	0	0	96	0	1	0.7
7	480	60	320	0	1	0	0	96	0	1	0.7
8	520	100	160	0	1	0	0	48	0	2	0.8
9	520	80	70	0	2	0	0	72	0	2	0.8
10	600	105	860	0	3	0	0	90	3	2	1
11	600	90	690	0	3	0	0	94	4	2	1
12	600	75	720	0	3	0	0	96	5	2	1
13	610	115	135	0	1	0	0	30	0	3	0.5
14	605	95	90	0	2	0	0	42	0	3	0.5
15	600	110	145	0	2	0	0	72	0	3	0.8
16	630	107	165	0	3	0	0	36	0	3	1
17	630	105	200	0	3	0	0	60	0	3	1
18	630	95	125	0	3	0	0	90	0	3	1
19	610	92	80	0	1	0	0	72	0	4	0.6
20	605	85	85	0	2	0	0	96	0	4	0.7
21	600	80	110	0	3	0	0	88	0	4	0.9
22	630	92	120	0	1	0	0	84	0	4	0.9
23	630	90	170	0	3	0	0	90	0	4	1
24	630	85	125	0	3	0	0	48	0	4	1
25	610	82	90	0	1	0	0	45	0	5	0.8
26	605	70	75	0	2	0	0	96	0	5	1
27	600	65	85	0	3	0	0	74	0	5	0.7
28	630	80	150	0	2	0	0	65	0	5	1
29	630	75	135	0	3	0	0	92	0	5	0.5
30	630	65	185	0	3	0	0	72	0	5	1

Hub/D	wMOGA	wMOGG	PMOGA
APOD	6	6	6
SPOD	0	6	0
1	0	3	0
2	0	4	0
3	0	3	0
4	0	3	0
5	0	3	0

cust	CummulativeDemand	ETDD	TDD
0	85	0	28
0	165	0	48
0	290	0	74
0	420	0	96
1	45	0	30
1	135	0	48
1	200	0	78
1	390	0	96
2	1220	0	96
3	2500	0	96
4	150	0	30
4	260	0	92
5	190	0	54
5	340	0	96
6	125	0	42
6	300	0	96
7	100	0	70
7	320	0	96
8	160	0	48
9	70	0	72
10	860	0	90
11	690	0	94
12	720	0	96
13	135	0	30
14	90	0	42
15	145	0	72
16	165	0	36
17	200	0	60
18	125	0	90
19	80	0	72
20	85	0	96
21	110	0	88
22	120	0	84
23	170	0	90
24	125	0	48
25	90	0	45
26	75	0	96
27	85	0	74
28	150	0	65
29	135	0	92
30	185	0	72

Data Set 22

Joint Multiple Trip Multiple Service with hubs TDVRSP
medium delivery restrictions, low demand/cap

cust	locationCoor		demand	WMOGA,WMOGG,PMOGA			eTDD	TDD	hub	tier	prior
0	400	75	400	1	3	2	0	96	0	0	1
1	400	90	200	2	2	4	0	96	0	0	1
2	400	105	800	2	2	4	0	96	1	0	1
3	500	90	1800	3	3	6	0	96	2	0	1
4	450	60	210	0	2	0	0	72	0	1	1
5	450	90	220	0	3	0	0	96	0	1	1
6	450	120	180	0	1	0	0	28	0	1	1
7	480	60	190	0	1	0	0	96	0	1	1
8	520	100	75	0	1	0	0	48	0	2	1
9	520	80	75	0	2	0	0	72	0	2	1
10	600	105	500	0	3	0	0	96	3	2	1
11	600	90	600	0	2	0	0	96	4	2	1
12	600	75	550	0	3	0	0	96	5	2	1
13	610	115	65	0	1	0	0	24	0	3	1
14	605	95	70	0	2	0	0	36	0	3	1
15	600	110	60	0	2	0	0	72	0	3	1
16	630	107	105	0	2	0	0	30	0	3	1
17	630	105	95	0	2	0	0	66	0	3	1
18	630	95	105	0	3	0	0	96	0	3	1
19	610	92	100	0	1	0	0	70	0	4	1
20	605	85	100	0	2	0	0	96	0	4	1
21	600	80	100	0	3	0	0	48	0	4	1
22	630	92	100	0	1	0	0	72	0	4	1
23	630	90	100	0	2	0	0	36	0	4	1
24	630	85	100	0	3	0	0	90	0	4	1
25	610	82	90	0	1	0	0	96	0	5	1
26	605	70	90	0	2	0	0	96	0	5	1
27	600	65	90	0	3	0	0	65	0	5	1
28	630	80	100	0	2	0	0	72	0	5	1
29	630	75	90	0	3	0	0	96	0	5	1
30	630	65	90	0	2	0	0	68	0	5	1

Hub/D	wMOGA	wMOGG	PMOGA
APOD	6	6	6
SPOD	0	6	0
1	0	3	0
2	0	4	0
3	0	3	0
4	0	3	0
5	0	3	0

cust	CummulativeDemand	ETDD	TDD
0	100	0	34
0	200	0	66
0	300	0	72
0	400	0	96
1	20	0	36
1	80	0	68
1	115	0	74
1	200	0	96
2	800	0	96
3	1800	0	96
4	100	0	33
4	210	0	68
5	110	0	66
5	220	0	96
6	180	0	28
7	90	0	60
7	190	0	96
8	75	0	45
9	75	0	64
10	500	0	96
11	600	0	96
12	550	0	96
13	65	0	20
14	70	0	40
15	60	0	60
16	105	0	36
17	95	0	72
18	105	0	96
19	100	0	64
20	100	0	88
21	100	0	42
22	100	0	54
23	100	0	33
24	100	0	84
25	90	0	88
26	90	0	96
27	90	0	72
28	100	0	68
29	90	0	90
30	90	0	72

Data Set 23

Joint Multiple Trip Multiple Service with hubs TDVRSP
medium delivery restrictions, medium demand/cap

cust	locationCoor		demand	WMOGA,WMOGG,PMOGA			eTDD	TDD	hub	tier	prior
0	400	75	350	1	3	2	0	96	0	0	1
1	400	90	350	2	2	4	0	96	0	0	1
2	400	105	970	2	2	4	0	96	1	0	1
3	500	90	2240	3	3	6	0	96	2	0	1
4	450	60	240	0	2	0	0	72	0	1	1
5	450	90	240	0	3	0	0	96	0	1	1
6	450	120	240	0	1	0	0	72	0	1	1
7	480	60	250	0	1	0	0	96	0	1	1
8	520	100	100	0	1	0	0	48	0	2	1
9	520	80	100	0	2	0	0	72	0	2	1
10	600	105	680	0	3	0	0	96	3	2	1
11	600	90	680	0	2	0	0	96	4	2	1
12	600	75	680	0	3	0	0	96	5	2	1
13	610	115	75	0	1	0	0	24	0	3	1
14	605	95	85	0	2	0	0	36	0	3	1
15	600	110	85	0	2	0	0	72	0	3	1
16	630	107	145	0	2	0	0	30	0	3	1
17	630	105	145	0	2	0	0	66	0	3	1
18	630	95	145	0	3	0	0	96	0	3	1
19	610	92	75	0	1	0	0	70	0	4	1
20	605	85	85	0	2	0	0	96	0	4	1
21	600	80	85	0	3	0	0	48	0	4	1
22	630	92	145	0	1	0	0	72	0	4	1
23	630	90	145	0	2	0	0	36	0	4	1
24	630	85	145	0	3	0	0	90	0	4	1
25	610	82	75	0	1	0	0	96	0	5	1
26	605	70	85	0	2	0	0	96	0	5	1
27	600	65	85	0	3	0	0	65	0	5	1
28	630	80	145	0	2	0	0	72	0	5	1
29	630	75	145	0	3	0	0	96	0	5	1
30	630	65	145	0	2	0	0	68	0	5	1

Hub/D	wMOGA	wMOGG	PMOGA
APOD	6	6	6
SPOD	0	6	0
1	0	3	0
2	0	4	0
3	0	3	0
4	0	3	0
5	0	3	0

cust	CummulativeDemand	ETDD	TDD
0	75	0	36
0	155	0	60
0	240	0	70
0	350	0	96
1	65	0	36
1	155	0	60
1	220	0	72
1	350	0	96
2	970	0	96
3	2240	0	96
4	110	0	36
4	240	0	72
5	130	0	60
5	240	0	96
6	120	0	36
6	240	0	72
7	100	0	60
7	250	0	96
8	100	0	48
9	100	0	72
10	680	0	96
11	680	0	96
12	680	0	96
13	75	0	24
14	85	0	36
15	85	0	72
16	145	0	30
17	145	0	66
18	145	0	96
19	75	0	70
20	85	0	96
21	85	0	48
22	145	0	72
23	145	0	36
24	145	0	90
25	75	0	96
26	85	0	96
27	85	0	65
28	145	0	72
29	145	0	96
30	145	0	68

Data Set 24

Joint Multiple Trip Multiple Service with hubs TDVRSP
medium delivery restrictions, high demand/cap

cust	locationCoor		demand	WMOGA,WMOGG,PMOGA			eTDD	TDD	hub	tier	prior
0	400	75	320	1	3	2	0	96	0	0	0.8
1	400	90	490	2	2	4	0	96	0	0	0.7
2	400	105	1140	2	2	4	0	96	1	0	1
3	500	90	2500	3	3	6	0	96	2	0	1
4	450	60	260	0	2	0	0	66	0	1	0.5
5	450	90	300	0	3	0	0	96	0	1	0.5
6	450	120	310	0	1	0	0	72	0	1	0.7
7	480	60	270	0	1	0	0	96	0	1	0.7
8	520	100	100	0	1	0	0	44	0	2	0.8
9	520	80	130	0	2	0	0	64	0	2	0.8
10	600	105	600	0	3	0	0	88	3	2	1
11	600	90	900	0	2	0	0	90	4	2	1
12	600	75	770	0	3	0	0	96	5	2	1
13	610	115	75	0	1	0	0	32	0	3	0.9
14	605	95	70	0	2	0	0	40	0	3	0.9
15	600	110	85	0	2	0	0	72	0	3	1
16	630	107	145	0	2	0	0	40	0	3	0.8
17	630	105	140	0	2	0	0	60	0	3	0.8
18	630	95	85	0	3	0	0	90	0	3	1
19	610	92	120	0	1	0	0	61	0	4	1
20	605	85	75	0	2	0	0	88	0	4	1
21	600	80	140	0	3	0	0	48	0	4	1
22	630	92	170	0	1	0	0	65	0	4	1
23	630	90	200	0	2	0	0	42	0	4	1
24	630	85	195	0	3	0	0	96	0	4	1
25	610	82	90	0	1	0	0	88	0	5	0.6
26	605	70	75	0	2	0	0	96	0	5	0.6
27	600	65	65	0	3	0	0	56	0	5	0.6
28	630	80	180	0	2	0	0	72	0	5	0.6
29	630	75	165	0	3	0	0	90	0	5	0.6
30	630	65	195	0	2	0	0	51	0	5	0.6

Hub/D	wMOGA	wMOGG	PMOGA
APOD	6	6	6
SPOD	0	6	0
1	0	3	0
2	0	4	0
3	0	3	0
4	0	3	0
5	0	3	0

cust	CummulativeDemand	ETDD	TDD
0	75	0	30
0	135	0	64
0	260	0	72
0	320	0	96
1	55	0	30
1	155	0	54
1	280	0	68
1	490	0	96
2	1140	0	96
3	2500	0	96
4	180	0	48
4	260	0	66
5	130	0	52
5	300	0	96
6	145	0	30
6	310	0	72
7	90	0	50
7	270	0	96
8	100	0	44
9	130	0	64
10	600	0	88
11	900	0	90
12	770	0	96
13	75	0	32
14	70	0	40
15	85	0	72
16	145	0	40
17	140	0	60
18	85	0	90
19	120	0	61
20	75	0	88
21	140	0	48
22	170	0	65
23	200	0	42
24	195	0	96
25	90	0	88
26	75	0	96
27	65	0	56
28	180	0	72
29	165	0	90
30	195	0	51

Data Set 25

Joint Multiple Trip Multiple Service with hubs TDVRSP
high delivery restrictions, low demand/cap

cust	locationCoor		demand	WMOGA,WMOGG,PMOGA			eTDD	TDD	hub	tier	prior
0	400	75	280	1	3	2	0	96	0	0	1
1	400	90	320	1	1	2	0	90	0	0	1
2	400	105	800	2	2	4	0	96	1	0	1
3	500	90	1800	3	3	6	0	96	2	0	1
4	450	60	250	0	2	0	0	68	0	1	1
5	450	90	150	0	3	0	0	88	0	1	1
6	450	120	180	0	1	0	0	68	0	1	1
7	480	60	220	0	1	0	0	96	0	1	1
8	520	100	100	0	1	0	0	45	0	2	1
9	520	80	100	0	1	0	0	64	0	2	1
10	600	105	500	0	3	0	0	72	3	2	1
11	600	90	500	0	2	0	0	70	4	2	1
12	600	75	600	0	3	0	0	92	5	2	1
13	610	115	65	0	1	0	0	20	0	3	1
14	605	95	70	0	2	0	0	30	0	3	1
15	600	110	70	0	1	0	0	64	0	3	1
16	630	107	95	0	2	0	0	40	0	3	1
17	630	105	95	0	1	0	0	80	0	3	1
18	630	95	105	0	3	0	0	60	0	3	1
19	610	92	55	0	1	0	0	36	0	4	1
20	605	85	60	0	1	0	0	82	0	4	1
21	600	80	60	0	1	0	0	45	0	4	1
22	630	92	95	0	1	0	0	66	0	4	1
23	630	90	115	0	2	0	0	88	0	4	1
24	630	85	115	0	3	0	0	48	0	4	1
25	610	82	80	0	1	0	0	40	0	5	1
26	605	70	80	0	2	0	0	90	0	5	1
27	600	65	80	0	1	0	0	54	0	5	1
28	630	80	80	0	2	0	0	66	0	5	1
29	630	75	80	0	3	0	0	48	0	5	1
30	630	65	100	0	1	0	0	78	0	5	1

Hub/D	wMOGA	wMOGG	PMOGA
APOD	6	6	6
SPOD	0	6	0
1	0	3	0
2	0	4	0
3	0	3	0
4	0	3	0
5	0	3	0

cust	CummulativeDemand	ETDD	TDD
0	69	0	20
0	142	0	48
0	210	0	72
0	280	0	96
1	80	0	18
1	160	0	36
1	235	0	74
1	320	0	90
2	800	0	96
3	1800	0	96
4	120	0	33
4	250	0	68
5	70	0	24
5	150	0	88
6	100	0	30
6	180	0	68
7	100	0	39
7	220	0	96
8	100	0	45
9	100	0	64
10	500	0	72
11	500	0	70
12	600	0	92
13	65	0	20
14	70	0	30
15	70	0	64
16	95	0	40
17	95	0	80
18	105	0	60
19	55	0	36
20	60	0	82
21	60	0	45
22	95	0	66
23	115	0	88
24	115	0	48
25	80	0	40
26	80	0	90
27	80	0	54
28	80	0	66
29	80	0	48
30	100	0	78

Data Set 26

Joint Multiple Trip Multiple Service with hubs TDVRSP
high delivery restrictions, medium demand/cap

cust	locationCoor		demand	WMOGA,WMOGG,PMOGA			eTDD	TDD	hub	tier	prior
0	400	75	300	1	3	2	0	96	0	0	1
1	400	90	400	1	1	2	0	96	0	0	1
2	400	105	970	2	2	4	0	96	1	0	1
3	500	90	2240	3	3	6	0	96	2	0	1
4	450	60	220	0	2	0	0	66	0	1	1
5	450	90	260	0	3	0	0	96	0	1	1
6	450	120	240	0	1	0	0	70	0	1	1
7	480	60	250	0	1	0	0	96	0	1	1
8	520	100	110	0	1	0	0	40	0	2	1
9	520	80	90	0	1	0	0	54	0	2	1
10	600	105	690	0	3	0	0	64	3	2	1
11	600	90	680	0	2	0	0	72	4	2	1
12	600	75	660	0	3	0	0	90	5	2	1
13	610	115	105	0	1	0	0	32	0	3	1
14	605	95	105	0	2	0	0	44	0	3	1
15	600	110	105	0	1	0	0	64	0	3	1
16	630	107	125	0	2	0	0	38	0	3	1
17	630	105	125	0	1	0	0	90	0	3	1
18	630	95	125	0	3	0	0	72	0	3	1
19	610	92	145	0	1	0	0	48	0	4	1
20	605	85	105	0	1	0	0	87	0	4	1
21	600	80	85	0	1	0	0	40	0	4	1
22	630	92	75	0	1	0	0	66	0	4	1
23	630	90	125	0	2	0	0	96	0	4	1
24	630	85	145	0	3	0	0	44	0	4	1
25	610	82	85	0	1	0	0	48	0	5	1
26	605	70	95	0	2	0	0	96	0	5	1
27	600	65	95	0	1	0	0	72	0	5	1
28	630	80	135	0	2	0	0	64	0	5	1
29	630	75	125	0	3	0	0	48	0	5	1
30	630	65	125	0	1	0	0	90	0	5	1

Hub/D	wMOGA	wMOGG	PMOGA
APOD	6	6	6
SPOD	0	6	0
1	0	3	0
2	0	4	0
3	0	3	0
4	0	3	0
5	0	3	0

cust	CummulativeDemand	ETDD	TDD
0	75	0	28
0	155	0	46
0	210	0	66
0	300	0	96
1	65	0	24
1	175	0	48
1	270	0	68
1	400	0	96
2	970	0	96
3	2140	0	96
4	110	0	30
4	220	0	66
5	130	0	36
5	260	0	96
6	130	0	42
6	240	0	70
7	100	0	48
7	250	0	96
8	110	0	40
9	90	0	54
10	690	0	64
11	680	0	72
12	560	0	90
13	105	0	32
14	105	0	44
15	105	0	64
16	125	0	38
17	125	0	90
18	125	0	72
19	145	0	48
20	105	0	87
21	85	0	40
22	75	0	66
23	125	0	96
24	145	0	44
25	75	0	48
26	85	0	96
27	85	0	72
28	105	0	64
29	105	0	48
30	105	0	90

Data Set 27

Joint Multiple Trip Multiple Service with hubs TDVRSP
high delivery restrictions, high demand/cap

cust	locationCoor		demand	WMOGA,WMOGG,PMOGA			eTDD	TDD	hub	tier	prior
0	400	75	390	1	3	2	0	96	0	0	0.8
1	400	90	420	1	1	2	0	96	0	0	0.8
2	400	105	1180	2	2	4	0	96	1	0	1
3	500	90	2500	3	3	6	0	96	2	0	1
4	450	60	280	0	2	0	0	72	0	1	1
5	450	90	300	0	3	0	0	96	0	1	1
6	450	120	310	0	1	0	0	72	0	1	1
7	480	60	290	0	1	0	0	96	0	1	1
8	520	100	120	0	1	0	0	48	0	2	1
9	520	80	110	0	1	0	0	72	0	2	1
10	600	105	760	0	3	0	0	72	3	2	0.8
11	600	90	790	0	2	0	0	70	4	2	1
12	600	75	720	0	3	0	0	96	5	2	0.9
13	610	115	95	0	1	0	0	24	0	3	0.5
14	605	95	90	0	2	0	0	36	0	3	0.5
15	600	110	105	0	1	0	0	72	0	3	0.5
16	630	107	165	0	2	0	0	30	0	3	0.5
17	630	105	160	0	1	0	0	84	0	3	0.5
18	630	95	145	0	3	0	0	68	0	3	0.5
19	610	92	100	0	1	0	0	40	0	4	1
20	605	85	85	0	1	0	0	96	0	4	1
21	600	80	120	0	1	0	0	48	0	4	1
22	630	92	150	0	1	0	0	72	0	4	1
23	630	90	180	0	2	0	0	90	0	4	1
24	630	85	155	0	3	0	0	48	0	4	1
25	610	82	80	0	1	0	0	45	0	5	0.7
26	605	70	85	0	2	0	0	96	0	5	0.7
27	600	65	75	0	1	0	0	65	0	5	0.7
28	630	80	160	0	2	0	0	72	0	5	0.7
29	630	75	145	0	3	0	0	42	0	5	0.7
30	630	65	175	0	1	0	0	96	0	5	0.7

Hub/D	wMOGA	wMOGG	PMOGA
APOD	6	6	6
SPOD	0	6	0
1	0	3	0
2	0	4	0
3	0	3	0
4	0	3	0
5	0	3	0

cust	CummulativeDemand	ETDD	TDD
0	85	0	24
0	165	0	42
0	260	0	70
0	390	0	96
1	65	0	20
1	155	0	40
1	230	0	72
1	420	0	96
2	1180	0	96
3	2500	0	96
4	130	0	36
4	280	0	72
5	150	0	28
5	300	0	96
6	125	0	36
6	310	0	72
7	100	0	42
7	290	0	96
8	120	0	48
9	110	0	72
10	760	0	72
11	790	0	70
12	720	0	96
13	95	0	24
14	90	0	36
15	105	0	72
16	165	0	30
17	160	0	84
18	145	0	68
19	100	0	40
20	85	0	96
21	120	0	48
22	150	0	72
23	180	0	90
24	155	0	48
25	80	0	45
26	85	0	96
27	75	0	65
28	160	0	72
29	145	0	42
30	175	0	96

Data Set 28

Air Force Multiple Trips Multiple Services with no hubs TDVRSP low delivery restrictions, high demand/cap with ETDD constraint

cust	locationCoor		demand	wMOGA	pMOGA	eTDD	TDD	priority
0	400	90	400	1	2	12	47	0.8
1	400	105	400	2	4	0	47	1
2	400	75	400	3	6	0	46	0.9
3	500	90	400	3	6	18	48	1
4	420	30	150	1	2	0	44	0.7
5	420	15	150	3	6	0	42	1
6	420	45	150	1	2	24	47	0.6
7	500	30	150	3	6	0	40	1

cust	CummulativeDemand	ETDD	TDD
0	100	12	20
0	197	12	32
0	299	12	39
0	400	12	47
1	105	0	18
1	215	0	32
1	310	0	42
1	400	0	47
2	97	0	18
2	194	0	26
2	303	0	39
2	400	0	46
3	212	18	32
3	292	18	42
3	400	18	48
4	75	0	15
4	150	0	44
5	58	0	18
5	150	0	42
6	150	24	47
7	96	0	30
7	150	0	40

Hub/Depot	WMOGA	WMOGG	PMOGA
APOD	6	Na	6

Data Set 29

Joint Multiple Trip Multiple Service with no hubs TDVRSP medium delivery restrictions, low demand/cap, and ETDD constraint

cust	locationCoor		demand	wMOGA	wMOGG	pMOGA	eTDD	TDD	prior
0	400	90	420	1	3	3	0	48	1
1	400	105	420	2	2	4	12	46	1
2	400	75	440	2	2	4	0	40	1
3	500	90	400	3	1	6	18	38	1
4	420	15	210	0	3	0	0	45	1
5	420	30	210	0	2	0	24	44	1
6	420	45	220	0	2	0	0	42	1
7	500	30	200	0	1	0	0	48	1

cust	CummulativeDemand	ETDD	TDD
0	71	0	12
0	242	0	30
0	340	0	35
0	420	0	48
1	254	12	32
1	333	12	36
1	420	12	46
2	120	0	18
2	215	0	24
2	290	0	32
2	440	0	40
3	190	18	27
3	298	18	34
3	400	18	38
4	50	0	12
4	110	0	24
4	165	0	36
4	210	0	45
5	210	24	48
6	68	0	18
6	140	0	28
6	180	0	36
6	220	0	42
7	64	0	20
7	142	0	32
7	177	0	36
7	200	0	48

Hub/Depot	WMOGA	WMOGG	PMOGA
APOD	6	Na	6
SPOD	0	6	0

Data Set 30

Joint Multiple Trip Multiple Service with hubs TDVRSP
high delivery restrictions, medium demand/cap, and ETDD constraints

cust	locationCoor		demand	WMOGA,WMOGG,PMOGA			eTDD	TDD	hub	tier	prior
0	400	75	300	1	3	2	12	96	0	0	1
1	400	90	400	1	1	2	0	96	0	0	1
2	400	105	970	2	2	4	0	96	1	0	1
3	500	90	2240	3	3	6	0	96	2	0	1
4	450	60	220	0	2	0	0	66	0	1	1
5	450	90	260	0	3	36	0	96	0	1	1
6	450	120	240	0	1	0	0	70	0	1	1
7	480	60	250	0	1	24	0	96	0	1	1
8	520	100	110	0	1	0	0	40	0	2	1
9	520	80	90	0	1	0	0	54	0	2	1
10	600	105	700	0	3	0	0	64	3	2	1
11	600	90	680	0	2	0	0	72	4	2	1
12	600	75	660	0	3	0	0	90	5	2	1
13	610	115	95	0	1	0	0	32	0	3	1
14	605	95	95	0	2	0	0	44	0	3	1
15	600	110	105	0	1	0	0	64	0	3	1
16	630	107	135	0	2	0	0	38	0	3	1
17	630	105	135	0	1	48	0	90	0	3	1
18	630	95	135	0	3	18	0	72	0	3	1
19	610	92	145	0	1	0	0	48	0	4	1
20	605	85	105	0	1	36	0	87	0	4	1
21	600	80	85	0	1	0	0	40	0	4	1
22	630	92	75	0	1	0	0	66	0	4	1
23	630	90	125	0	2	24	0	96	0	4	1
24	630	85	145	0	3	0	0	44	0	4	1
25	610	82	85	0	1	0	0	48	0	5	1
26	605	70	95	0	2	0	0	96	0	5	1
27	600	65	95	0	1	24	0	72	0	5	1
28	630	80	135	0	2	0	0	64	0	5	1
29	630	75	125	0	3	0	0	48	0	5	1
30	630	65	125	0	1	48	0	90	0	5	1

Hub/D	wMOGA	wMOGG	PMOGA
APOD	3	6	6
SPOD	0	6	0
1	0	3	0
2	0	4	0
3	0	3	0
4	0	3	0
5	0	3	0

cust	CummulativeDemand	ETDD	TDD
0	75	12	28
0	155	12	46
0	210	12	66
0	300	12	96
1	65	0	24
1	175	0	48
1	270	0	68
1	400	0	96
2	970	0	96
3	2240	0	96
4	110	0	30
4	220	0	66
5	130	12	96
5	260	0	96
6	130	0	42
6	240	0	70
7	100	24	54
7	250	24	96
8	110	0	40
9	90	0	54
10	800	0	64
11	680	0	72
12	560	0	90
13	95	0	32
14	105	0	44
15	105	0	64
16	165	0	38
17	165	48	90
18	165	18	72
19	145	0	48
20	105	36	87
21	85	0	40
22	75	0	66
23	125	24	96
24	145	0	44
25	75	0	48
26	85	0	96
27	85	24	72
28	105	0	64
29	105	0	48
30	105	48	90

Data Set 31

Air Force Multiple Trip Multiple Service with no hubs TDVRSP medium delivery restrictions, low demand/cap, and MTW constraints

cust	locationCoor		demand	wMOGA	pMOGA	eTDD	TDD	prior
0	400	90	315	1	3	0	48	1
1	400	105	315	2	4	0	46	1
2	400	75	315	2	4	0	40	1
3	500	90	315	3	6	0	38	1
4	420	15	105	3	6	0	45	1
5	420	30	105	2	4	0	44	1
6	420	45	105	2	4	0	42	1
7	500	30	105	1	1	0	48	1

cust	CummulativeDemand	ETDD	TDD
0	71	0	12
0	142	0	30
0	240	0	35
0	315	0	48
1	77	0	15
1	154	0	32
1	233	0	36
1	315	0	46
2	80	0	18
2	155	0	24
2	260	0	32
2	315	0	40
3	83	0	20
3	170	0	27
3	238	0	34
3	315	0	38
4	25	0	12
4	45	0	24
4	75	0	36
4	105	0	45
5	55	0	30
5	87	0	36
5	105	0	44
6	28	0	18
6	50	0	28
6	80	0	36
6	105	0	42
7	24	0	20
7	105	0	48

Hub/Depot	WMOGA	WMOGG	PMOGA
APOD	6	Na	6

Customer ID	AC-time windows	
3	48	54
6	18	24
7	24	36

Data Set 32

Joint Multiple Trip Multiple Service with no hubs TDVRSP high delivery restrictions, medium demand/cap, and MTW constraints

cust	locationCoor		demand	wMOGA	wMOGG	pMOGA	eTDD	TDD	prior
0	400	75	490	1	3	2	0	46	1
1	400	90	490	1	1	2	0	48	1
2	400	105	500	2	2	4	0	42	1
3	500	90	520	3	1	6	0	48	1
4	420	15	250	0	1	0	0	48	1
5	420	30	240	0	2	0	0	46	1
6	420	45	260	0	3	0	0	44	1
7	500	30	250	0	1	0	0	48	1

cust	CummulativeDemand	ETDD	TDD
0	102	0	10
0	244	0	22
0	374	0	35
0	490	0	46
1	125	0	14
1	251	0	22
1	375	0	34
1	510	0	48
2	99	0	18
2	262	0	23
2	364	0	32
2	500	0	42
3	82	0	14
3	275	0	21
3	373	0	33
3	520	0	48
4	88	0	16
4	122	0	24
4	250	0	48
5	50	0	12
5	110	0	20
5	180	0	30
5	270	0	46
6	192	0	36
6	270	0	44
7	95	0	14
7	155	0	24
7	190	0	36
7	270	0	48

Customer ID	AC-time windows		G- time windows	
1	48	60	48	60
4	24	36	24	36
6	6	18	6	18

Hub/Depot	WMOGA	WMOGG	PMOGA
APOD	**4**	**Na**	**6**
SPOD	**0**	**4**	**0**

Data Set 33

Joint Multiple Trip Multiple Service with hubs TDVRSP low delivery restrictions, high demand/cap, and MTW constraints

cust	locationCoor		demand	WMOGA,WMOGG,PMOGA			eTDD	TDD	hub	tier	prior
0	400	75	420	1	3	2	0	96	0	0	0.9
1	400	90	390	2	3	4	0	96	0	0	0.8
2	400	105	1180	3	3	6	0	96	1	0	1
3	500	90	2500	3	3	6	0	96	2	0	1
4	450	60	260	0	2	0	0	92	0	1	0.8
5	450	90	340	0	2	0	0	96	0	1	0.8
6	450	120	300	0	1	0	0	96	0	1	0.7
7	480	60	320	0	1	0	0	96	0	1	0.7
8	520	100	160	0	1	0	0	48	0	2	0.8
9	520	80	70	0	2	0	0	72	0	2	0.8
10	600	105	860	0	3	0	0	90	3	2	1
11	600	90	690	0	3	0	0	94	4	2	1
12	600	75	720	0	3	0	0	96	5	2	1
13	610	115	135	0	1	0	0	30	0	3	0.5
14	605	95	90	0	2	0	0	42	0	3	0.5
15	600	110	145	0	2	0	0	72	0	3	0.8
16	630	107	165	0	3	0	0	36	0	3	1
17	630	105	200	0	3	0	0	60	0	3	1
18	630	95	125	0	3	0	0	90	0	3	1
19	610	92	80	0	1	0	0	72	0	4	0.6
20	605	85	85	0	2	0	0	96	0	4	0.7
21	600	80	110	0	3	0	0	88	0	4	0.9
22	630	92	120	0	1	0	0	84	0	4	0.9
23	630	90	170	0	3	0	0	90	0	4	1
24	630	85	125	0	3	0	0	48	0	4	1
25	610	82	90	0	1	0	0	45	0	5	0.8
26	605	70	75	0	2	0	0	96	0	5	1
27	600	65	85	0	3	0	0	74	0	5	0.7
28	630	80	150	0	2	0	0	65	0	5	1
29	630	75	135	0	3	0	0	92	0	5	0.5
30	630	65	185	0	3	0	0	72	0	5	1

Hub/D	wMOGA	wMOGG	PMOGA
APOD	6	6	6
SPOD	0	6	0
1	0	3	0
2	0	4	0
3	0	3	0
4	0	3	0
5	0	3	0

cust	CummulativeDemand	ETDD	TDD
0	85	0	28
0	165	0	48
0	290	0	74
0	420	0	96
1	45	0	30
1	135	0	48
1	200	0	78
1	390	0	96
2	1220	0	96
3	2500	0	96
4	150	0	30
4	260	0	92
5	190	0	54
5	340	0	96
6	125	0	42
6	300	0	96
7	100	0	70
7	320	0	96
8	160	0	48
9	70	0	72
10	860	0	90
11	690	0	94
12	720	0	96
13	135	0	30
14	90	0	42
15	145	0	72
16	165	0	36
17	200	0	60
18	125	0	90
19	80	0	72
20	85	0	96
21	110	0	88
22	120	0	84
23	170	0	90
24	125	0	48
25	90	0	45
26	75	0	96
27	85	0	74
28	150	0	65
29	135	0	92
30	185	0	72

Customer ID	AC- time windows		G-time windows	
0	28	40	28	40
4			48	72
6			18	24
10			54	62
Hub 3	54	62	54	62

Data Set 34

Air Force Multiple Trip Multiple Service with no hubs TDVRSP high delivery restrictions, medium demand/cap, with ETDD and MTW constraints

cust	locationCoor		demand	wMOGA	pMOGA	eTDD	TDD	prior
0	400	75	380	1	2	12	46	1
1	400	90	380	1	2	0	48	1
2	400	105	380	2	4	6	42	1
3	500	90	380	3	6	0	48	1
4	420	15	120	1	2	0	48	1
5	420	30	120	2	4	18	46	1
6	420	45	120	3	6	0	44	1
7	500	30	120	1	1	0	48	1

cust	CummulativeDemand	ETDD	TDD
0	174	12	22
0	274	12	35
0	380	12	46
1	85	0	14
1	181	0	22
1	275	0	34
1	380	0	48
2	71	6	18
2	182	6	23
2	264	6	32
2	380	6	42
3	82	0	14
3	175	0	21
3	263	0	33
3	380	0	48
4	28	0	16
4	52	0	22
4	98	0	34
4	120	0	48
5	20	0	12
5	50	0	20
5	80	18	30
5	120	18	46
6	30	0	18
6	92	0	36
6	120	0	44
7	25	0	14
7	55	0	24
7	120	0	48

Customer ID	AC- time windows	
3	48	54
6	18	24
7	24	36

Hub/Depot	WMOGA	WMOGG	PMOGA
APOD	4	Na	6

Data Set 35

Joint Multiple Trip Multiple Service with no hubs TDVRSP
low delivery restrictions, high demand/cap, with ETDD and MTW constraints

cust	locationCoor		demand	wMOGA	wMOGG	pMOGA	eTDD	TDD	prior
0	400	90	550	1	3	2	0	47	1
1	400	105	500	2	2	4	0	47	0.7
2	400	75	550	3	3	6	6	46	1
3	500	90	600	3	2	6	0	48	0.9
4	420	30	300	0	1	0	10	44	1
5	420	15	250	0	3	0	0	42	1
6	420	45	300	0	1	0	0	47	0.8
7	500	30	250	0	3	0	18	40	1.0

cust	CummulativeDemand	ETDD	TDD
0	100	0	20
0	550	0	47
1	125	0	18
1	285	0	32
1	410	0	42
1	500	0	47
2	100	6	18
2	294	6	26
2	403	6	39
2	550	6	46
3	152	0	23
3	287	0	32
3	492	0	42
3	600	0	48
4	75	10	18
4	300	10	44
5	58	0	18
5	250	0	42
6	184	0	33
6	300	0	47
7	150	18	30
7	250	18	40

Customer	AC-time window		G-time window	
0	24	32	24	32
5			48	60
6			12	18

Hub/Depot	WMOGA	WMOGG	PMOGA
APOD	**6**	**Na**	**6**
SPOD	**0**	**6**	**0**

Data Set 36

Joint Multiple Trip Multiple Service with hubs TDVRSP
medium delivery restrictions, low demand/cap, with ETDD and MTW constraints

cust	locationCoor		demand	WMOGA,WMOGG,PMOGA			eTDD	TDD	hub	tier	prior
0	400	75	400	1	3	2	0	96	0	0	1
1	400	90	200	2	2	4	0	96	0	0	1
2	400	105	800	2	2	4	0	96	1	0	1
3	500	90	1800	3	3	6	0	96	2	0	1
4	450	60	210	0	2	0	0	72	0	1	1
5	450	90	220	0	3	0	0	96	0	1	1
6	450	120	180	0	1	0	0	28	0	1	1
7	480	60	190	0	1	0	0	96	0	1	1
8	520	100	75	0	1	0	0	48	0	2	1
9	520	80	75	0	2	0	48	72	0	2	1
10	600	105	500	0	3	0	0	96	3	2	1
11	600	90	600	0	2	0	0	96	4	2	1
12	600	75	550	0	3	0	0	96	5	2	1
13	610	115	65	0	1	0	0	24	0	3	1
14	605	95	70	0	2	0	0	36	0	3	1
15	600	110	60	0	2	0	0	72	0	3	1
16	630	107	105	0	2	0	0	30	0	3	1
17	630	105	95	0	2	0	18	72	0	3	1
18	630	95	105	0	3	0	18	96	0	3	1
19	610	92	100	0	1	0	36	70	0	4	1
20	605	85	100	0	2	0	36	96	0	4	1
21	600	80	100	0	3	0	0	48	0	4	1
22	630	92	100	0	1	0	0	72	0	4	1
23	630	90	100	0	2	0	0	36	0	4	1
24	630	85	100	0	3	0	36	90	0	4	1
25	610	82	90	0	1	0	24	96	0	5	1
26	605	70	90	0	2	0	24	96	0	5	1
27	600	65	90	0	3	0	24	65	0	5	1
28	630	80	100	0	2	0	24	72	0	5	1
29	630	75	90	0	3	0	24	96	0	5	1
30	630	65	90	0	2	0	24	68	0	5	1

Hub/Depot	WMOGA	WMOGG	PMOGA
APOD	6	6	6
SPOD	0	6	0
1	0	2	0
2	0	3	0
3	0	3	0
4	0	2	0
5	0	3	0

cust	CummulativeDemand	ETDD	TDD
0	100	0	34
0	200	0	66
0	300	0	72
0	400	0	96
1	20	0	36
1	80	0	68
1	115	0	74
1	200	0	96
2	800	0	96
3	1800	0	96
4	100	0	33
4	210	0	68
5	110	0	66
5	220	0	96
6	180	0	28
7	90	0	60
7	190	0	96
8	75	0	45
9	75	48	64
10	500	0	96
11	600	0	96
12	550	0	96
13	65	0	20
14	70	0	40
15	60	0	60
16	105	0	36
17	95	18	72
18	105	18	96
19	100	36	64
20	100	36	88
21	100	0	42
22	100	0	54
23	100	0	33
24	100	36	84
25	90	24	88
26	90	24	96
27	90	24	72
28	100	24	68
29	90	24	90
30	90	24	72

cust	AC-time window	
APOD	24	30
APOD	48	54
APOD	72	78
APOD	96	102

Data Set 37

Joint Multiple Trip Multiple Service with hubs TDVRSP
low delivery restrictions, low demand/cap, ETDD, MTW, storage, RL constraints

cust	locationCoor		dem	wMOGA, wMOGG, pMOGA			eTDD	TDD	hub	tier	prior	storage	ACfuel	Gfuel
0	400	75	300	1	3	2	0	96	0	0	1	0	0	300
1	400	90	300	2	3	4	0	96	0	0	1	0	0	300
2	400	105	800	3	3	6	0	96	1	0	1	200	2000	500
3	500	90	1800	3	3	6	0	96	2	0	1	300	8000	600
4	450	60	200	0	2	0	0	96	0	1	1	0	0	200
5	450	90	200	0	2	0	0	96	0	1	1	0	0	200
6	450	120	200	0	1	0	36	96	0	1	1	0	0	200
7	480	60	200	0	1	0	24	96	0	1	1	0	0	200
8	520	100	75	0	1	0	0	48	0	2	1	0	0	50
9	520	80	75	0	2	0	0	72	0	2	1	0	0	50
10	600	105	550	0	3	0	0	96	3	2	1	0	0	200
11	600	90	550	0	3	0	0	96	4	2	1	0	0	200
12	600	75	550	0	3	0	0	96	5	2	1	0	0	200
13	610	115	65	0	1	0	0	24	0	3	1	0	0	50
14	605	95	70	0	2	0	0	36	0	3	1	0	0	50
15	600	110	70	0	2	0	0	72	0	3	1	0	0	50
16	630	107	115	0	3	0	0	30	0	3	1	0	0	50
17	630	105	115	0	3	0	0	66	0	3	1	0	0	50
18	630	95	115	0	3	0	0	96	0	3	1	0	0	50
19	610	92	65	0	1	0	24	70	0	4	1	0	0	50
20	605	85	70	0	2	0	24	96	0	4	1	0	0	50
21	600	80	70	0	3	0	24	96	0	4	1	0	0	50
22	630	92	115	0	1	0	24	96	0	4	1	0	0	50
23	630	90	115	0	3	0	24	36	0	4	1	0	0	50
24	630	85	115	0	3	0	24	90	0	4	1	0	0	50
25	610	82	65	0	1	0	48	96	0	5	1	0	0	50
26	605	70	70	0	2	0	48	96	0	5	1	0	0	50
27	600	65	70	0	3	0	48	96	0	5	1	0	0	50
28	630	80	115	0	2	0	48	72	0	5	1	0	0	50
29	630	75	115	0	3	0	48	96	0	5	1	0	0	50
30	630	65	115	0	3	0	48	72	0	5	1	0	0	50

Hub/D	wMOGA	wMOGG	PMOGA
APOD	6	8	6
SPOD	0	8	0
1	0	4	0
2	0	4	0
3	0	4	0
4	0	4	0
5	0	4	0

cust	CummulativeDemand	ETDD	TDD
0	69	0	36
0	142	0	60
0	219	0	84
0	300	0	96
1	70	0	36
1	150	0	60
1	215	0	84
1	300	0	96
2	800	0	96
3	1800	0	96
4	90	0	36
4	200	0	96
5	110	0	60
5	200	0	96
6	200	36	96
7	80	24	60
7	200	24	96
8	75	0	48
9	75	0	72
10	550	0	96
11	550	0	96
12	550	0	96
13	65	0	24
14	70	0	36
15	70	0	72
16	115	0	30
17	115	0	66
18	115	0	96
19	65	0	70
20	70	0	96
21	70	0	96
22	115	0	96
23	115	0	36
24	115	0	90
25	65	0	96
26	70	0	96
27	70	0	96
28	115	0	72
29	115	0	96
30	115	0	72

cust	AC-timewindows		G- time windows	
APOD	24	30	0	0
APOD	48	54	0	0
APOD	72	78	0	0
2	24	30	0	0
2	48	54	0	0
2	72	78	0	0

Data Set 38

Joint Multiple Trip Multiple Service with hubs TDVRSP
Med delivery restrictions, med demand/cap, ETDD, MTW, storage, RL constraints

cust	coor location		dem	wMOGA,wMOGG,pMOGA			eTDD	TDD	hub	tier	prior	storage	Afuel	Gfuel
0	400	75	350	1	3	2	0	96	0	0	1	0	0	400
1	400	90	350	2	2	4	24	96	0	0	1	0	0	400
2	400	105	970	2	2	4	0	96	1	0	1	200	2000	600
3	500	90	2240	3	3	6	0	96	2	0	1	300	8000	500
4	420	15	240	0	2	0	0	72	0	1	1	0	0	200
5	420	30	240	0	3	0	0	96	0	1	1	0	0	200
6	420	45	240	0	1	0	0	72	0	1	1	0	0	200
7	500	30	250	0	1	0	0	96	0	1	1	0	0	200
8	520	100	100	0	1	0	24	48	0	2	1	0	0	50
9	520	80	100	0	2	0	36	72	0	2	1	0	0	50
10	600	105	680	0	3	0	0	96	3	2	1	100	0	200
11	600	90	680	0	2	0	0	96	4	2	1	100	0	200
12	600	75	680	0	3	0	0	96	5	2	1	100	0	200
13	610	115	75	0	1	0	18	30	0	3	1	0	0	50
14	605	95	85	0	2	0	18	36	0	3	1	0	0	50
15	600	110	85	0	2	0	18	72	0	3	1	0	0	50
16	630	107	145	0	2	0	18	30	0	3	1	0	0	50
17	630	105	145	0	2	0	18	66	0	3	1	0	0	50
18	630	95	145	0	3	0	18	96	0	3	1	0	0	50
19	610	92	75	0	1	0	0	70	0	4	1	0	0	50
20	605	85	85	0	2	0	0	96	0	4	1	0	0	50
21	600	80	85	0	3	0	0	48	0	4	1	0	0	50
22	630	92	145	0	1	0	0	72	0	4	1	0	0	50
23	630	90	145	0	2	0	0	36	0	4	1	0	0	50
24	630	85	145	0	3	0	0	90	0	4	1	0	0	50
25	610	82	75	0	1	0	36	96	0	5	1	0	0	50
26	605	70	85	0	2	0	36	96	0	5	1	0	0	50
27	600	65	85	0	3	0	36	65	0	5	1	0	0	50
28	630	80	145	0	2	0	36	72	0	5	1	0	0	50
29	630	75	145	0	3	0	36	96	0	5	1	0	0	50
30	630	65	145	0	2	0	36	68	0	5	1	0	0	50

Hub/D	wMOGA	wMOGG	PMOGA
APOD	6	6	6
SPOD	0	6	0
1	0	2	0
2	0	3	0
3	0	3	0
4	0	2	0
5	0	3	0

cust	CummulativeDemand	ETDD	TDD
0	75	0	36
0	155	0	60
0	240	0	70
0	350	0	96
1	155	24	60
1	220	24	72
1	350	24	96
2	970	0	96
3	2240	0	96
4	110	0	36
4	240	0	72
5	130	0	60
5	240	0	96
6	120	0	36
6	240	0	72
7	100	0	60
7	250	0	96
8	100	24	48
9	100	24	72
10	680	0	96
11	680	0	96
12	680	0	96
13	75	18	30
14	85	18	36
15	85	18	72
16	145	18	30
17	145	18	66
18	145	18	96
19	75	0	70
20	85	0	96
21	85	0	48
22	145	0	72
23	145	0	36
24	145	0	90
25	75	36	96
26	85	36	96
27	85	36	65
28	145	36	72
29	145	36	96
30	145	36	68

cust	AC-timewindows		G- time windows	
Hub 2	36	48		
Hub 2	72	84		
Hub 4			24	48
3	36	48	0	0
3	72	84	0	0
4			36	48
5			48	60
11			24	48

Data Set 39

Joint Multiple Trip Multiple Service with hubs TDVRSP
high delivery restrictions, high demand/cap, ETDD, MTW, storage, RL constraints

cust	Coor location		dem	wMOGA,wMOGG,pMOGA			eTDD	TDD	hub	tier	prior	storage	Afuel	Gfuel
0	400	75	390	1	3	2	10	96	0	0	0.8	0	0	400
1	400	90	420	1	1	2	0	96	0	0	0.8	0	0	400
2	400	105	1180	2	2	4	0	96	1	0	1	200	2000	400
3	500	90	2500	3	3	6	0	96	2	0	1	200	8000	500
4	420	15	280	0	2	0	0	72	0	1	1	0	0	100
5	420	30	300	0	3	0	0	96	0	1	1	0	0	100
6	420	45	310	0	1	0	20	72	0	1	1	0	0	100
7	500	30	290	0	1	0	24	96	0	1	1	0	0	100
8	520	100	120	0	1	0	0	48	0	2	1	0	0	50
9	520	80	110	0	1	0	0	72	0	2	1	0	0	50
10	600	105	760	0	3	0	0	72	3	2	0.8	100	0	200
11	600	90	790	0	2	0	0	70	4	2	1	100	0	200
12	600	75	720	0	3	0	0	96	5	2	0.9	100	0	200
13	610	115	95	0	1	0	18	40	0	3	0.5	0	0	50
14	605	95	90	0	2	0	18	36	0	3	0.5	0	0	50
15	600	110	105	0	1	0	18	72	0	3	0.5	0	0	50
16	630	107	165	0	2	0	18	48	0	3	0.5	0	0	50
17	630	105	160	0	1	0	18	84	0	3	0.5	0	0	50
18	630	95	145	0	3	0	18	68	0	3	0.5	0	0	50
19	610	92	100	0	1	0	12	40	0	4	1	0	0	50
20	605	85	85	0	1	0	12	96	0	4	1	0	0	50
21	600	80	120	0	1	0	12	48	0	4	1	0	0	50
22	630	92	150	0	1	0	12	72	0	4	1	0	0	50
23	630	90	180	0	2	0	12	90	0	4	1	0	0	50
24	630	85	155	0	3	0	12	48	0	4	1	0	0	50
25	610	82	80	0	1	0	0	45	0	5	0.7	0	0	50
26	605	70	85	0	2	0	0	96	0	5	0.7	0	0	50
27	600	65	75	0	1	0	0	65	0	5	0.7	0	0	50
28	630	80	160	0	2	0	0	72	0	5	0.7	0	0	50
29	630	75	145	0	3	0	0	42	0	5	0.7	0	0	50
30	630	65	175	0	1	0	0	96	0	5	0.7	0	0	50

Hub/D	wMOGA	wMOGG	PMOGA
APOD	3	6	6
SPOD	0	6	0
1	0	2	0
2	0	3	0
3	0	3	0
4	0	2	0
5	0	3	0

cust	CummulativeDemand	ETDD	TDD
0	85	10	24
0	165	10	42
0	260	10	70
0	390	10	96
1	65	0	20
1	155	0	40
1	230	0	72
1	420	0	96
2	1180	0	96
3	2500	0	96
4	130	0	36
4	280	0	72
5	150	0	28
5	300	0	96
6	125	20	36
6	310	20	72
7	100	24	42
7	290	24	96
8	120	0	48
9	110	0	72
10	760	0	72
11	790	0	70
12	720	0	96
13	95	18	40
14	90	18	36
15	105	18	72
16	165	18	48
17	160	18	84
18	145	18	68
19	100	12	40
20	85	12	96
21	120	12	48
22	150	12	72
23	180	12	90
24	155	12	48
25	80	0	45
26	85	0	96
27	75	0	65
28	160	0	72
29	145	0	42
30	175	0	96

cust	AC-timewindows		G- time windows	
APOD	30	36		
Hub 3			18	30
0	84	96		
1	48	60		
10			18	30

Bibliography

Aggarwal, A.K., M. Oblak, and R.R. Vemugantis. “A Heuristic Solution Procedure For Multicommodity Integer Flows,” Computers and Operations Research: Vol. 22, No.10, 1075-1087 (1995).

Arnow, D. and G. Weiss. Introduction to Programming Using An Object-Oriented Approach. Addison-Wesley, 2000.

Assad, A.A. and B.L. Golden. Vehicle Routing: Methods and Studies. Amsterdam: North Holland, 1988.

Augerat P., J.M. Belenguer, E. Benavent, A. Corberán and D. Naddef. “Separating Capacity Constraints In The CVRP Using Tabu Search,” European Journal of Operational Research: Vol.106, No.2-3, 546-557(1998).

Baker, B. M. and J. Sheasby. “Extensions To The Generalised Assignment Heuristic For Vehicle Routing,” European Journal of Operational Research: Vol. 119, 147 – 157 (1999).

Balinski, M.L. and R.E. Quandt. “On an Integer Program for the Delivery Program” Operations Research: 12, 300-304 (1964).

Barbarosoglu, G. and D. Ozgur. “A Tabu Search Algorithm For The Vehicle Routing Problem,” Computers and Operations Research: Vol. 26, 255 – 270 (1999).

Barnard, T. and H. Neill. Mathematical Groups. Berkshire: Cox and Wyman Ltd, 1996.

Barnes, J. W. and W. B. Carlton. “Solving the Traveling – Salesman Problem with Time Windows using Tabu Search,” IIE Transactions: Vol. 28, 617 –629 (1996).

Barnes, J. W., B. Colletti, and D.L. Neway. “Group Theory And Transition Matrices For Conjugative Move Disciplines In Multiple TSP’s With Known Solution Structure,” technical paper to be published, University of Texas at Austin, 2000.

Barnes, J. W. and B.W. Colletti. “Local Search Structure In The Symmetric Traveling Salesperson Problem Under A General Class Of Rearrangement Neighborhoods,” technical paper, the University of Texas at Austin, 2000.

Barnes, Dallis. “Velocity Management,” Quatermaster Professional Bulletin: Autumn, 1996.

Battiti, R. and G. Tecchiollo. “The Reactive Tabu Search,” ORSA Journal on Computing: Vol. 6 No. 2, 126-140 (1994).

Baumslag, B. and B. Chandler. Group Theory: Schaum's Outline Series. MacGraw Hill, 1968.

Bertsimas, D. J. and D. Simchi-Levi. "A New Generation Of Vehicle Routing Research: Robust Algorithms, Addressing Uncertainty," Operations Research: Vol. 44, No. 2, 286 – 304 (1996).

Bodin, L. D. "Twenty Years Of Routing And Scheduling," Operations Research: Vol. 38, 571 – 579 (1999).

Bodin, L., B. Golden, A. Assad, and M. Ball. "Routing and Scheduling of Vehicles and Crews; The State of The Art," Computers and Operations Research, 10, (1983).

Bodin, L. and L. Berman, "Routing and Scheduling of School Buses by Computer," Transportation Sciences: 13, 113-129 (1979).

Brandão, J. and A. Mercer. "A Tabu Search Algorithm For The Multi-Trip Vehicle Routing And Scheduling Problem," European Journal of Operational Research: Vol. 100, No. 1, 180-191 (1997).

Brooks, Roger. "Position Paper on the Theater Distribution Management System (TDMS) Initiative" Headquarters, United States Air Force, Washington DC, 26 May 2000.

Brooks, Roger. "Theater Distribution Management System Briefing", Washington: HQ USAF, 26 May 2000.

Brown, Darin T. Routing Unmanned Aerial Vehicles While Considering General Restricted Operating Zones, Masters Thesis, Air Force Institute of Technology, 2001

Calhoun, Kevin M. A Tabu Search For Scheduling And Rescheduling Combat Aircraft, Masters Thesis, Air Force Institute of Technology, 2000.

Capehart, S. R. A Tabu Search Metaheuristic for the Air Refueling Tanker Assignment Problem, Masters Thesis, Air Force Institute of Technology, 2000.

Carlton, W. B. A Tabu Search Approach to the General Vehicle Routing Problem, Ph.D. Dissertation, The University of Texas at Austin, 1995.

Chiang, W. C. and R. A. Russell. "A Reactive Tabu Search Metaheuristic For The Vehicle Routing Problem With Time Windows," INFORMS Journal of Computing: Vol. 9, No. 4, 417–430 (1995).

Christofides, H. "Vehicle Routing and Scheduling," Presented at the 12th International Symposium on Mathematical Programming, 1985.

Christofides, H., A. Mingozzi, and P. Toth. "Exact Algorithms for the Vehicle Routing Problem, Based on Spanning Tree and Shortest Path Relaxations," Mathematical Programming: 20, 255-282 (1981).

Christofides, N., A. Mignozzi, and P. Toth. "The Vehicle Routing Problem," Combinatorial Optimization: Chicester: Wiley, 1979.

Clarke, G. and J.W. Wright. "Scheduling Vehicles from a Central Depot to a Number of Delivery Points," Operations Research: 12, 568-581 (1964).

Colletti, B. W. Group Theory and Metaheuristics. Ph.D. dissertation, The University of Texas at Austin, 1999.

Colletti, B.W. Information provided directly to this research effort, (2001,2002).

Colletti, B. W. and J.W. Barnes. Group Theory and Metaheuristic Search Neighborhoods. ORP99-02, The Graduate Program in Operations Research and Industrial Engineering, The University of Texas at Austin, 1999a.

Colletti, B. W. and J.W. Barnes. "Linearity in the Traveling Salesman Problem," Submitted to Applied Mathematics Letters, 1999b.

Colletti, B. W. and J.W. Barnes. "Using Group Theory to Efficiently Compute Move Values for Routing Problems," Submitted to Journal of Heuristics, 1999c.

Colletti, B. W., J.W. Barnes, and S. Dokov. "A Note on Characterizing the k-OPT Neighborhood Via Group Theory," Journal of Heuristics: Vol. 5, 51-55 (1999).

Cordeau, JF, M. Gendreau, and G. Laporte. "A Tabu Search Heuristic For Periodic And Multi-Depot Vehicle Routing Problems," Networks: 30, 105-119 (1997).

Cullenbine, Christopher A. A Tabu Search Approach To The Weapons Assignment Model, Masters Thesis, Air Force Institute of Technology, 2000.

Department of the Chairman of the Joint Chiefs of Staff. Joint Tactics, Techniques, and Procedures for Joint Theater Distribution. Joint Publication 4-01.4. Washington: CJCS, 22 August 2000.

Department of the Chairman of the Joint Chiefs of Staff. Department of Defense Dictionary of Military and Associated Terms. Joint Publication 1-02. Washington: CJCS, 9 September1993.

Department of the Chairman of the Joint Chiefs of Staff. Joint Tactics, Techniques, and Procedures for Joint Reception, Staging, Onward Movement, and Integration. Joint Publication 4-01.8. Washington: CJCS, 13 June 2000.

Department of the Chairman of the Joint Chiefs of Staff. Joint Tactics, Techniques, and Procedures for Movement Control. Joint Publication 4-01.3. Washington: CJCS, 21 June 1996.

Department of the Army. Reception, Staging, Onward Movement, and Integration. Field Manual 100-17-3. Washington: USA, 17 March 1999.

Department of the Army. Theater Distribution. Field Manual 100-10-1. Washington: USA, 1 October 1999.

Department of the Army. Movement Control. Field Manual 55-10. Washington: USA, 9 February 1999.

Department of the Air Force. Air Mobility Operations. Air Force Doctrine Document 2-6. Washington: 25 June 1999.

Department of the Air Force. Air Lift Operations. Air Force Doctrine Document 2-6.1. Washington: 13 November 1999.

Department of the Air Force. Combat Support. Air Force Doctrine Document 2-6. Washington: 22 November 1999.

Desrochers, M., C. V. Jones, J. K. Lenstra, M. W. P. Savelsbergh, and L. Stougie. "Towards A Model And Algorithm Management System For Vehicle Routing And Scheduling Problems," Draft, November 1998.

Desrouchers, M.C., J.K. Lenstra, and M.W.P. Savelsbergh. "A Classification Scheme for Vehicle Routing and Scheduling Problems," European Journal of Operations Research: Vol. 46, 322 – 332 (1990).

Desrouchers, M.C., J. Desrosiers, and M.M. Solomon. "A New Optimization Algorithm for The Vehicle Routing Problem with Time Windows," Operations Research: 342-353, (1992).

Desaulniers G., J. Lavigne, and F. Soumis. "Multi-Depot Vehicle Scheduling Problems With Time Windows And Waiting Costs," European Journal of Operational Research: Vol. 111, No. 3, 479 – 494 (1998).

--------. "ELIST briefing," Military Traffic Management Command, Transportation Engineering Agency, Washington.

Fisher, M.L. and R. Jaikumar. " A Generalized Assignment Heurisitic for Vehicle Routing," Networks: 11, 109-124 (1981).

Fraleigh, John B. A First Course in Abstract Algebra, Addison-Wesley, Reading MA, 1976.

Gendreau, M., G. Laporte, and A. Hertz. "New Insertion and Post Optimization Procedures for the Traveling Salesman Problem," Operations Research: 40, 6, 1086-1094 (1992).

Gendreau, M., A. Hertz, G. Laporte, and M. Stan. "A Generalized Insertion Heuristic For The Traveling Salesman Problem With Time Windows," Operations Research: Vol. 43, 330 – 335 (1998).

Gendreau, M., F. Guertin, J. Y. Potvin, and É. Taillard. "Parallel Tabu Search For Real-Time Vehicle Routing And Dispatching," Transportation Science: Vol. 33, No. 4, 381 – 390 (1999).

Gendreau, M., G. Laporte, C. Musaraganyi and É. D. Taillard. "A Tabu Search Heuristic For The Heterogeneous Fleet Vehicle Routing Problem," Computers & Operations Research: Vol. 26, No. 12, 1153-1173 (1999).

Gendreau, M., A. Hertz, and G. Laporte. " A Tabu Search Heuristic for the Vehicle Routing Problem," Management Science: 40, 1276 – 1290 (1994).

Ghiani, G. and G. Improta. "An Efficient Transformation Of The Generalized Vehicle Routing Problem," European Journal of Operational Research, Vol. 122, 11 – 17 (2000).

Gillett, B.E. and L.R. Miller. "A Heuristic Algorithm for the Vehicle Dispatch Problem," Operations Research: 22, 240-349 (1974).

Glover, F. and M. Laguna. Tabu Search, Boston: Kluwar Academic Publishers, 1997.

Grover, L.K.. "Local Search And The Local Structure Of NP-Complete Problems," Operations Research Letters:12, 235-243 (1992).

Golden, B. L., G. Laporte, and É. D. Taillard. "An Adaptive Memory Heuristic For A Class Of Vehicle Routing Problems With Minmax Objective," Computers & Operations Research: Vol. 24, No. 5, 445 – 452 (1997).

Golden, B., T. Magnanti, and H. Nguyen. "Implementing Vehicle Routing Algorithms," Networks: 7, 113-148 (1977).

Gomory, Ralph E. "Outline of an Algorithm for Integer Solutions to Linear Programs," Bulletin of the American Mathematical Society: 64, 275-278 (1958).

Gomory, Ralph E. "An Algorithm to Integer Solutions to Linear Programs," In Recent Advances in Mathematical Programming, R. Graves, P.Wolfe (eds), McGraw Hill, New York, 269-302 (1963)

Hachicha, M., M. J. Hodgson, G. Laporte and F. Semet "Heuristics For The Multi-Vehicle Covering Tour Problem," Computers & Operations Research: Vol. 27, No. 1, 29-42 (2000).

Hall, S. N.. A Group Theoretic Tabu Search Approach to the Traveling Salesman Problem. Masters Thesis, Air Force Institute of Technology, 2000.

Hamber, B., "T.LoaDS Abbreviated Systems Architecture", Proceedings of the 2001 Winter Simulation Conference, 2001.

Harder, R.W. A Jump Search Enhanced Reactive Tabu Search In A Universal Vehicle Router Applied To Routing Unmanned Aerial Vehicles, Master's Thesis, Air Force Institute of Technology, 2000.

Hodgson T.J., O. Spivey, T. Trainor, and M.K. Williams. "The Deployment Analysis Network Tool Extended (DANTE)," Military Operations Research: Vol. 6, No.3, 5-18, 2001.

Hong, S. and Y. B. Park. "A Heuristic For Bi-Objective Vehicle Routing With Time Window Constraints," International Journal of Production Economics: Vol. 62, 249 – 258 (1999).

HQ USAF/ILA. "Theater Distribution Management System Requirements and Feasibility Study", 15 January, 2002.

Irnich, S. "A Multi-Depot Pickup And Delivery Problem With A Single Hub And Heterogeneous Vehicles," European Journal of Operational Research: Vol. 122, No. 2, 310-328 (2000).

Isaacs, I.M. Algebra, Brooks/Cole Publishing, Pacific Grove CA, 1994.

Kaminski, Paul G. "The Revolution in Defense Logistics", Keynote address of the USAD for Acquisition and Technology, Alexandria VA, 31 October 1995.

Kelly, J.P. and J. Xu. " A Set Partitioning Based Heuristic For The Vehicle Routing Problem," INFORMS Journal on Computing: Vol. 11, No. 2, 161-172 (1999).

Kim, J.U. and Y.D. Kim. "A Decomposition Approach to a Multi-Period Vehicle Scheduling Problem," Omega, The International Journal of Management Science: 27, 421-430 (1999).

Kinney, Gary W. A Hybrid Jump Search And Tabu Search Metaheuristic For The Unmanned Aerial Vehicle (UAV) Routing Problem, Masters Thesis, Air Force Institute of Technology, 2000.

Laporte, G., M. Gendreau, J.Y. Potvin and F. Semet. "Classical And Modern Heuristics For The Vehicle Routing Problem," International Transactions in Operational Research: 7(4-5), 285-300 (2000).

Laporte, G.. "The Vehicle Routing Problem: An Overview of Exact and Approximate Algorithms" European Journal of Operational Research: 59, 345-358 (1992).

Laporte, G., H. Mercure, and Y. Nobert. "An Exact Algorithm for the Asymmetrical Capacitated Vehicle Routing Problem," Networks: 16, 33-46 (1986).

Laporte, G.H., Y. Nobert, and M. Deroshers. "Optimal Routing under Capacity and Distance Restrictions," Operations Research: 33, 1050-1073 (1985).

Lin, S. "Computer Solutions of the Traveling Salesman Problem", Bell System Technical Journal: 44, 2245-2269 (1965).

Liu, F.H. F. and S. Y. Shen. "An Overview Of A Heuristic For Vehicle Routing Problem With Time Windows," Computers & Industrial Engineering: Vol. 37, 331 – 334 (1999).

Liu, F.H. F. and S. Y. Shen. "A Route-Neighborhood-Based Metaheuristic For Vehicle Routing Problem With Time Windows," European Journal of Operational Research: Vol. 118, 485 – 504 (1999).

Nanry, W. P. and J. W. Barnes. "Solving The Pickup And Delivery Problem With Time Windows Using Reactive Tabu Search," Transportation Research Part B: Vol. 34, 07 – 121 (2000).

Oglesby, Philip. Requirement for C-130 Aircraft in the Intratheater Korean Scenario. Masters Thesis, Air Force Institute of Technology, 1996

Osman, I.H. "Meta Strategy Simulated Annealing and Tabu Search," Annals of Operations Research: 40:1, 421-452 (1993).

O'Rourke, Kevin P. Dynamic Unmanned Aerial Vehicle (UAV) Routing With A Java-Encoded Reactive Tabu Search Metaheuristic, Master's thesis, Air Force Institute of Technology, 1999

Potvin, J. Y., T. Kervahut, B. L. Garcia, and J. M. Rousseau. "The Vehicle Routing Problem With Time Windows - Part I : Tabu Search," INFORMS Journal of Computing: Vol. 8, 158 – 164 (1996).

Powell, W. B. and T. A. Carvalho. "Dynamic Control Of Multicommodity Fleet Management Problems," European Journal of Operational Research: Vol. 98, No.3,522-541 (1997).

Rego, C., and C. Roucairol. " A Parallel Tabu Search Algorithm Using Ejection Chains for the Vehicle Routing Problem," In: Osman, I.H. and Kelly, J.P. Meta-Heuristics, Theory and Applications. Boston: Kluwer, 1996.

Rochat, Y. and E.D. Taillard. "Probablistic Diversifiaction and Intensification in Local Search for Vehicle Routing," Journal of Heuristics: 1, 147-167 (1995).

Rodríguez, P., M. Nussbaum, R. Baeza, G. León, M. Sepúlveda, and A. Cobián. "Using Global Search Heuristics For The Capacity Vehicle Routing Problem," Computers & Operation Research: Vol. 25, No. 5, 407 – 417 (1998).

Rose, J.S. A Course On Group Theory. New York: Dover Publications Inc, 1978.

Ryer, D. M. Implementation of the Metaheuristic Tabu Search in Route Selection for Mobility Analysis Support System, Masters Thesis, Air Force Institute of Technology, 1999.

Ryan, J.L., T.G. Bailey, J.T. Moore, and W.B. Carlton. "Unmanned Aerial Vehicle (UAV) Route Selection Using Reactive Tabu Search," Military Operations Research: Vol. 4 No. 3, 5-24 (1999).

Sagan, Bruce E.. The Symmetric Group. Pacific Grove: Wadsworth and Brooks/Cole, 1991.

Salhi S. and M. Sari. "A Multi-Level Composite Heuristic For The Multi-Depot Vehicle Fleet Mix Problem," European Journal of Operational Research: Vol. 103, No. 1, 95-112 (1997).

Savelsbergh, M.H.P. "Local Search in Routing Problems with Time Windows," Annals of Operations Research: 14, 285-305 (1985).

Schatz, Rowayne A. Theater Airlift Lessons from Kosovo. Air University, Maxwell AFB, 10 July 2000.

Schrady, David. Combatant Logistics Command and Control for the Joint Force Commander. Navy report, 1999.

Shalikashvili, John M.. "Joint Vision 2010," Joint Force Quarterly: Summer 1996.

Solomon, M.M.. "Algorithms for the Vehicle Routing and Scheduling Problem with Time Window Constraints," Operational Research:35, 254-265 (1987).

Solomon, M.M., E.K. Baker, and J.R. Schaffer. "Vehicle Routing and Scheduling Problems with Time Window Constraints: Efficient Implementations of Solution Improvement Procedures," in Vehicle Routing: Methods and Studies, B.L. Golden and A.A. Assad (eds) Amsterdam: North Holland, 85-105, (1988).

Solomon, M.M. and J. Desrosiers. " Time Window Constrained Routing and Scheduling Problems," Transportation Sciences: 22, 1-13 (1988).

Taillard, E.D.. "Parallel Iterative Search Methods for the Vehicle Routing Problem," Networks: 23, 661-673 (1993).

Taillard, E.D., G. Laporte, and M. Gendreau. "Vehicle Routing and Multiple Use of Vehicles," Journal of the Operations Research Society: 47, 1065-1070 (1996).

Toth, P. and D. Vigo. "Granular Tabu Search," Working Paper, DEIS, University of Bologna, 1998.

United States General Accounting Office. DoD's Mobility Requirements: Value of Intratheater Lift Analysis can be Enhanced. Washington: GAO/NSIAD-98-53, 30 January 1998.

Wiley, V. D. The Aerial Refueling Problem – A Doctoral Dissertation Proposal. The University of Texas at Austin, 2000.

Wiley, V. D. A Step By Step Users Guide To Implementing The Group Class In Java. User Manual, The University of Texas at Austin, 2000.

Wiley, V. D.. The Symmetric Group Class Users Manual For Partitioning And Ordering Problems, Users Manual, The University of Texas at Austin, 2000.

Wiley, V.D. The Aerial Fleet Refueling Problem. Ph.D. Dissertation, The University of Texas at Austin, 2001.

Xu, J. and J.P. Kelly. "A Network Flow Based Tabu Search Heuristic for the Vehicle Routing Problem," Transportation Science: 30, 379-393 (1996).

Vita

John Crino was born and raised in the small town of Silver Creek, NY. He graduated from Silver Creek High School in 1983 and was honored as the first person from his town to attend the United States Military Academy at West Point. He graduated from West Point in 1987, was commissioned as an Infantry Officer, and attended the Infantry Officer Basic Course and Ranger School. His first tour was with the 25th Infantry Division, Schofield Barracks, HI, where he led an infantry rifle platoon and was a company executive officer. During that tour, he represented the 25th ID in the National Best Ranger Competition, where he and his buddy, SSGT Jeff Anderson, finished 4th place, the best ever finish by a non-Ranger Regiment team.

After completing the Infantry Officer Advance Course and upon promotion to Captain, he was assigned to the 82nd Airborne Division, Fort Bragg NC, where he served as an Assistant Battalion Operations Officer and Company Commander. He commanded Alpha Company 1-505th Parachute Infantry Regiment and led them on a number of deployments, which include Panama, Germany, and JRTC.

In 1996, John earned his Masters of Science degree in Mathematical and Computer Sciences from the Colorado School of Mines. His thesis was titled: *Measuring the Efficiency of US Army Combat Units: An Application of Data Envelopment Analysis*.

After graduation, he was assigned to the Office of The Deputy Chief of Staff for Personnel, Pentagon, where he served as the Strength Management System Redesign Program Manager and Manpower Analyst. As the SMSR Program Manager, he directed a $20 million effort to redesign the Army's manpower decision support systems.

In 1999, MAJ John Crino attended the Air Force Institute of Technology Operations Research Ph.D. program, where he majored in optimization and minored in simulation. His research efforts concentrated in the application of group theory to tabu search when solving difficult combinatorial optimization problems. He developed a generalized model that solves theater distribution vehicle routing and scheduling problems.

John's next assignment is with the Office of the Chief of Staff of the Army, Program Analysis and Evaluation Directorate, as the Army's program analyst for the acquisition of air defense and space hardware and equipment.

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 074-0188*

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE**	**3. DATES COVERED** *(From – To)*
18-06-2002	**Doctoral Dissertation**	April 2001 – May 2002

4. TITLE AND SUBTITLE	
A GROUP THEORETIC TABU SEARCH METHODOLOGY FOR SOLVING THE THEATER DISTRIBUTION VEHICLE ROUTING AND SCHEDULING PROBLEM	**5a. CONTRACT NUMBER**
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	**5d. PROJECT NUMBER** ENR # 2000-061
Crino, John Richard, Major, US Army	**5e. TASK NUMBER**
	5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S)	**8. PERFORMING ORGANIZATION REPORT NUMBER**
Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/EN) 2950 P Street, Building 640, WPAFB OH 45433-7765	AFIT/DS/ENS/02-02
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	**10. SPONSOR/MONITOR'S ACRONYM(S)**
AFOSR/NM, Attn: Maj Juan Vasquez, 801 North Randolph St., Rm. 732 Arlington VA 22203-1977 phone:(703) 696-8431 DSN: 426-8431 FAX: (703) 696-8450; E-mail: juan.vasquez@afosr.af.mil	**11. SPONSOR/MONITOR'S REPORT NUMBER(S)**

12. DISTRIBUTION/AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES

14. ABSTRACT
The application of Group Theory to Tabu Search is a new and exciting field of research. This dissertation applies and extends some of Colletti's (1999) seminal work in group theory and metaheuristics in order to solve the theater distribution vehicle routing and scheduling problem (TDVRSP).

This research produced a robust, efficient, effective and flexible generalized theater distribution model that prescribes the routing and scheduling of multi-modal theater transportation assets to provide economically efficient time definite delivery of cargo to customers. In doing so, advances are provided in the field of group theoretic tabu search and its application to difficult combinatorial optimization problems, e.g., the multiple trip multiple services vehicle routing and scheduling problem with hubs and other defining constraints.

15. SUBJECT TERMS
Group Theory, Symmetric Group on n-letters, Metaheuristics, Tabu Search, Vehicle Routing Problems, Vehicle Scheduling Problems, Vehicle Routing and Scheduling Problems, Theater Distribution

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** Dr. James T. Moore, AFIT/ENS
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			**19b. TELEPHONE NUMBER** *(Include area code)*
U	U	U	UU	366	(937) 255-6565, ext 4337; e-mail: James.Moore@afit.edu